HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta/Osasto — Fakultet/Sektion — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Mathematics and Statistics

Tekijä — Författare — Author
Ensio Suonperä

Työn nimi — Arbetets titel — Title

Shearlet-based projection to wavefront set prior with convolutional neural network

Oppiaine — Läroämne — Subject
Mathematics

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's Thesis	October 2019	53

Tiivistelmä — Referat — Abstract

The motivation for the methods developed in this thesis rises from solving the severely ill-posed inverse problem of limited angle computed tomography. Breast tomosynthesis provides an example where the inner structure of the breast should be reconstructed from a very limited measurement angle. Some parts of the boundaries of the structure can be recovered from the X-ray measurements and others can not. These are referred to as visible and invisible boundaries. For parallel beam measurement geometry directions of visible and invisible boundaries can be deduced from the measurement angles. This motivates the usage of the concept of wavefront set. Roughly speaking, a wavefront set contains boundary points and their directions. The definition of wavefront set is based on Fourier analysis, but its characterization with the decay properties of functions called shearlets is used in this thesis. Shearlets are functions based on changing resolution, orientation, and position of certain generating functions. The theoretical part of this thesis focuses on studying this connection between shearlets and wavefront sets.

This thesis applies neural networks to the limited angle CT problem since neural networks have become state-of-the-art in many computer vision tasks and achieved impressive performance in inverse problems related to imaging. Neural networks are compositions of multiple simple functions, typically alternating linear functions and some element-wise non-linearities. They are trained to learn values for a huge amount of parameters to approximate the desired relation between input and output spaces. Neural networks are very flexible function approximators, but high dimensional optimization of parameters from data makes them hard to interpret. Convolutional neural networks (CNN) are the ones that succeed in tasks with image-like inputs. U-Net is a CNN architecture with very good properties, like learning useful parameters form considerably small data sets. This thesis provides two U-Net based CNN methods for solving limited angle CT problems. The main focus is on method projecting model-based reconstructions such that the projections have the desired wavefront sets. The guiding principle of this projector network is that it should not change reconstruction already projected to the given wavefront set. Another network estimates the invisible part of the wavefront set from the visible one. Few different data sets are simulated to train and evaluate these methods and performance on real data is also tested. A combination of the wavefront set estimator and the projector networks were used to postprocess model-based reconstructions. The fact this postprocessing has two steps increases the interpretability and the control over the processes performed by neural networks. This postprocessing increased the quality of reconstructions significantly and quality was even better when the true wavefront set was given for the projector as a prior.

Avainsanat — Nyckelord — Keywords
inverse problems, limited angle tomography, wavefront set, shearlets, convolutional neural networks, U-Net

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta/Osasto — Fakultet/Sektion — Faculty	Laitos — Institution — Department
Matemaattis-luonnontieteellinen	Matematiikan ja tilastotieteen laitos

Tekijä — Författare — Author
Ensio Suonperä

Työn nimi — Arbetets titel — Title

Shearlet-based projection to wavefront set prior with convolutional neural network

Oppiaine — Läroämne — Subject		
Matematiikka		
Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Pro gradu -tutkielma	Lokakuu 2019	53

Tiivistelmä — Referat — Abstract

Tässä työssä kehittettiin ratkaisumenetelmä rajoitetun kulman tomografiaan, joka on huonosti asetettu inversio-ongelma. Esimerkki tällaisesta ongelmasta on rinnan tomosynteesi, missä rinnan sisäinen rakenne on tarkoitus rekonstruoida todella rajoitetun kulman mittauksesta. Osa rakenteen reunoista pystytään selvittämään röntgenmittauksista, mutta ei kaikkia. Näitä kutsutaan näkyviksi ja näkymättömiksi reunoiksi. Yhdensuuntaisten säteiden mittausgeometriassa näkyvien ja näkymättömien reunojen suunnat voi päätellä mittaussuunnista. Tämä motivoi käyttämään aaltorintamajoukon käsitettä. Karkeasti ottaen aaltorintamajoukko sisältää reunapisteet ja niiden suunnat. Aaltorintamajoukon määritelmä perustuu Fourier analyysiin mutta tässä työssä käytetään sen karakterisointia shearlet:eiksi kutsuttujen funktioiden suppenemisominaisuuksien perusteella. Shearlet:it ovat funktioita, jotka perustuvat tiettyjen generoivien funktioiden resoluution, suuntautumisen ja sijainnin muuttamiseen. Tutkielman teoreettinen osio keskittyy tämän shearlet:ien ja aaltorintamajoukon välisen yhteyden tarkasteluun.

Tässä tutkielmassa sovelletaan neuroverkkoja rajoitetun kulman tomografiaan, koska niistä on tullut alan parhaimmistoa monien konenäkötehtävien ratkaisussa ja niillä on saavutettu vaikuttavia tuloksia kuvantamiseen liittyvissä inversio-ongelmissa. Neuroverkot ovat useista yksinkertaisista funktioista koostuvia yhdistettyjä funktioita, joissa tyypillisesti vuorottelevat lineaariset ja komponenteittain sovellettevat epälineaariset funktiot. Ne koulutetaan oppimaan sopivat arvot suurelle määrälle parametreja siten että ne approksimoivat haluttua riippuvuutta kahden avaruuden välillä. Neuroverkot kykenevät approksimoimaan hyvin laajaa joukkoa funktioita, mutta korkeaulotteinen parametrien optimointi datasta tekee niiden tulkinnan hankalaksi. Kuvankaltaisten signaalien käsittelyssä mestyvät erityisesti konvoluutioneuroverkot. U-Net on konvoluutioneuroverkkotyyppi, jolla on erityisen hyviä ominaisuuksia, kuten kyky oppia hyödylliset parametrit varsin pienestä koulutusjoukosta. Tämä tutkielma esittelee kaksi U-Net:iin perustuvaa konvoluutioneuroverkkomenetelmää rajoitetun kulman tomografia -ongelmien ratkaisemiseksi. Päähuomio on menetelmässä, joka projisoi perinteisellä menetelmällä luotuja rekonstruktioita siten että projektioilla on vaadittu aaltorintamajoukko. Tämän projektion pääperiaateen mukaan sen tulee säilyttää jo projisoitu rekonstruktio mahdollisimman samanlaisena projisoidessa uudestaan. Toinen menetelmä arvioi aaltorintamajoukon näkymätöntä osaa sen näkyvän osan perusteella. Näiden menetelmien kouluttamiseksi ja testaamiseksi simuloitiin erilaisia aineistoja. Suoriutumista tarkastellaan myös oikealla röntgenmittauksella. Aaltorintamajoukon arvioija- ja projektoijaneuroverkkoa käytettiin perinteisen menetelmän rekonstruktioiden jälkikäsittelyyn. Kaksivaiheinen jälkikäsittely lisää neuroverkkojen oppimien muutosten tulkittavuutta ja hallintaa. Tehty jälkikäsittely paransi rekonstruktioiden laatua huomattavasti ja laatu oli vieläkin parempi, kun projektioneuroverkolle annettiin tarkasteltavan kohteen aaltorintamajoukko esitietona arvioidun aaltorintamajoukon sijasta.

Avainsanat — Nyckelord — Keywords
inversio-ongelmat, rajoitetun kulman tomografia, aaltorintamajoukko, shearlets, konvoluutioneuroverkot, U-Net

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Shearlet-based projection to wavefront set prior with convolutional neural network

University of Helsinki

Ensio Suonperä

October 14, 2019

Acknowledgments

I want to thank Samuli Siltanen and Tatiana Bubba for supervising the thesis and Samuli also for introducing the intriguing field of inverse problems to me. I would not have got this far without the Finnish education system, my family and friends, especially ones at the University of Helsinki.

Contents

1 Introduction

X-ray tomography, also known as *computed tomography* (CT), was introduced by Allan Cormack and Godfrey Hounsfield (Nobel laureates) in the 1970s [10] and it is widely applied in non-destructive testing [14] and medical imaging [11]. In X-ray tomography, the inner structure of the target is reconstructed from indirect measurement data. This data is obtained by taking X-ray projection images of the target from several directions. These X-ray projection images contain only indirect information about the target, how much X-rays attenuated along their paths through the target. Such problems, where the object needs to be recovered from indirect measurements, are called *inverse problems.* Inverse problems are typically *ill-posed*, which mean they fail to fulfill at least one of the *well-posed* properties known as Hadamard conditions [37]:

H_1: **Existence.** There should be at least one solution.

H_2: **Uniqueness.** There should be at most one solution.

H_3: **Stability.** The solution must depend continuously on the data.

Note that if the stability condition H_3 holds, small changes to data affect only small changes to the solution. Especially failing of the stability condition H_3 typically makes an ill-posed problem hard to solve. However, there is a standard method called *filtered back-projection* (FBP) that yields good reconstructions for CT problems when complete data is available. Idealized complete tomographic data contains measurements from all directions, and dense enough sampling represents it quite well in a practical situation. If only limited data is available, the corresponding tomography problem is severely ill-posed. The data might be limited due to the need to reduce radiation dose or restrictions from the measurement setting.

This thesis focuses on developing strategies to solve a *limited angle computed tomography* problem. In this problem, the X-ray projection images of the target are measured only from directions in a limited angular range. Breast tomosynthesis provides an example where the angular range is very limited, 50° and 60° for some measurement devices [47] or even less [35] and with sparse angles. The limited angle CT problem is severely ill-posed [13] and only some part of the boundaries of the target can be recovered reliably from the measurement. Mathematically this can be formalized using microlocal analysis and the notion of *wavefront set.* Roughly, the wavefront set of the target contains boundary points and their directions. That part of the wavefront set, which can be recovered reliably from the measurement, is called *visible* and the other part *invisible.* For parallel beam measurement geometry directions of visible and invisible wavefront sets can be deduced from the measurement directions [39, 17]. Although the invisible part can not be directly recovered from the measurement, natural objects can be regular enough to deduce it from the visible part. This task requires appropriate prior information about the distribution of the measured objects. However, it can be hard to define the prior information explicitly. This is where data-based methods can be helpful.

Supervised learning is an area of *machine learning*, where a relation between input and output spaces is approximated by learning from given (input, output) pairs, which are called training data. *Artificial neural networks* are functions originally developed to mimic human brains introduced in 1943 by McCulloch and Pitts [33], but later similarity with brains has become less important. Neural networks are a very flexible function class by universal approximation theorem [24] and therefore a powerful framework for supervised learning [19]. Machine learning using neural networks is referred to as *deep learning.* Due to the increased amount of training data and computing resources, deep learning methods have become state-of-the-art in many computer vision tasks, including classification, object detection, and segmentation [43, 18]. Different approaches using neural networks have also achieved impressive performance solving inverse problems [3]. The role of neural network in these approaches has been post-processing, removing artifacts and noise from reconstruction achieved by traditional methods [22, 23, 26] or to replace proximal operators [1, 34]. The use of *convolutional neural networks* (CNNs), which are based on the discrete convolution operation, is

common for the deep learning methods for computer vision tasks and imaging-related inverse problems mentioned earlier.

There are approaches using deep learning to solve limited angle tomography problem [46, 22, 8, 36]. In the article [22], different CNN architectures performance on post-process FBP reconstructions was evaluated. U-Net type multiresolution architectures were found to perform better than single resolution architecture. The best performance was obtained with a method learning in the *wavelet* domain instead of the image domain. Wavelets are functions based on changing resolution and position of certain generating functions [12]. Even more impressive performance was obtained in the paper [8] with a hybrid method in the *shearlet* domain. Shearlets are similar to wavelets, but also the orientation of the generating functions is varied [31]. This makes shearlets better for describing data with directional information, like edges in the images. The hybrid method in [8] used directionality of shearlets to divide the shearlet domain into visible and invisible parts and inferring only invisible part by deep learning. It also used architecture similar to U-Net but added residual connections to consecutive layers at each scale. The visible part was reconstructed with an advanced model-based method that achieved better results than FBP. The described methods were evaluated with data measured in angular ranges of at least 100°, but the focus of this thesis is on more limited angle cases, 50° and 80°. The method presented in [36] achieved impressive performance in problem with 48° angular range using an approach quite different from the others described here. This method replaced proximal operators of an iterative minimization algorithm with CNNs and also used a prior mask.

The fact that the wavefront set of the target can be divided into visible and invisible parts in limited angle tomography serves as a basis for the methods developed for this thesis. In step 1 a model-based reconstruction is computed to extract the visible part. These reconstructions were achieved with the positivity-constrained total variation regularization. Next, in step 2, the reconstructions from step 1 were transformed into the shearlet domain, because the decay properties of shearlet coefficients give a practical way to characterize the wavefront set. This method splits post-processing with deep learning to two subtasks, estimation of the invisible wavefront set and projecting the reconstruction of step 1 to the space where the wavefront set of each element equals the estimated wavefront set. Developing methods to complete these subtasks is the contribution of this thesis. The chosen network architectures were residual modifications of U-Net and highly motivated by the choices in articles [22] and [8]. The novel idea was to regularize training of the projection network penalizing the difference between outputs of projecting the input once and twice. The operation that the network performs is called *projection* according to this property. The true target was chosen to guide, how the projection should change the reconstruction such that it has the desired wavefront set. The procedure to obtain reconstruction with developed methods is summarised as:

Step 1: Obtain a model-based reconstruction $\tilde{\mathbf{f}}$ from the measurement.

Step 2: Extract the visible part of the wavefront set from the reconstruction $\tilde{\mathbf{f}}$.

Step 3: Use a neural network to estimate the invisible part of the wavefront set from the visible one.

Step 4: Use a neural network to project the reconstruction $\tilde{\mathbf{f}}$ such that the projection has the estimated wavefront set.

Even if the performance of the projection network was tested with a post-processing approach, it was designed to be also used in iterative algorithms. Some iterative minimization algorithms have steps applying proximal operator and projection is a special case of proximal operator [9].

The structure of this thesis is described next. The theoretical background of the limited angle tomography problem and the method provided to face it is presented in section 2. It includes the concept of wavefront set, a brief introduction to shearlets, and the theory providing a connection between them.

Section 3 is devoted to introducing neural networks and practical aspects of their use to solve supervised machine learning tasks. Both of the sections 2 and 3 provides background for the developed methods that are described in section 4. In section 5, different experimental scenarios are specified, and the corresponding results are provided. The thesis is concluded with a brief discussion about future perspectives in section 6.

2 Theoretical background

2.1 X-ray tomography model

Let start by considering the model of the target object and then describe the measurement process. This thesis focuses on 2D tomography, where attenuation properties are reconstructed for a slice of an object. The entire 3D object could be reconstructed one 2D slice at a time. A slice of the object lies in a compact set $\Omega \subset \mathbb{R}^2$. There are different ways of modeling an attenuation function $f : \Omega \to \mathbb{R}$. Usual prior information about the target is that the attenuation values are non-negative because the target is only attenuating X-rays, not emitting them. For the theoretical setting f can be extended to $\mathbb{R}^2$ such that $f(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathbb{R}^2 \setminus \Omega$. A considerable amount of analysis is built on Lebesgue spaces. For $1 \leq p < \infty$ the Lebesgue space $L^p(\mathbb{R}^d)$ is defined as

$$\|f\|_{L^p} = \left(\int |f(\boldsymbol{x})|^p d\boldsymbol{x} \right)^{1/p} < \infty$$

and $L^\infty(\mathbb{R}^d)$ as functions such that

$$\|f\|_{L^\infty} = \inf\{C \geq 0 : \ |f(x)| \leq C \text{ for almost every } x \in \mathbb{R}^d\} < \infty.$$

In practice, the attenuation function is bounded and compactly supported. Therefore it holds $f \in L^\infty(\mathbb{R}^2)$ and also $f \in L^p(\mathbb{R}^2)$ for $1 \leq p < \infty$, since

$$\|f\|_{L^p} = \left(\int_\Omega |f(\boldsymbol{x})|^p d\boldsymbol{x} \right)^{1/p} \leq \left(\int_\Omega \|f\|_{L^\infty}^p d\boldsymbol{x} \right)^{1/p} = \|f\|_{L^\infty} m(\Omega) < \infty.$$

Although L^p-spaces provide a useful theoretical framework, there are more realistic models for measurement objects since L^p-spaces contain very irregular functions. One mathematical way to impose regularity is to consider spaces of k-times continuously differentiable functions $C^k(\mathbb{R}^d)$. Natural images basically consist of smooth regions separated by edges, which motivates the following definition for the class of functions describing natural objects [31].

Definition 1. The *class* $\mathcal{E}^2(\mathbb{R}^2)$ *of cartoon-like images* is the set of functions $f : \mathbb{R}^2 \to \mathbb{R}$ of the form

$$f = \sum_i f_i \mathbb{1}_{B_i},$$

where $B_i \subset \Omega$ are sets with piecewise C^2-smooth boundaries and $f_i \in C^2(\mathbb{R}^2)$ are functions supported in Ω. With rescaling and translation the domain Ω can be set to be $[0, 1]^2$.

In tomography measurement, the X-rays travel through the domain Ω along straight lines. The intensity of X-rays at the source I_0 is known from the calibration, and the intensity of X-rays traveled to detector I_1 is measured. The amount of attenuation of an X-ray along its path from source to the detector is determined by how strongly attenuating regions it travels through and how long it travels through each specific region. The mathematical model for the process based on the physics of X-rays is

$$\int_{L(\theta,r)} f(\boldsymbol{x}) dS(\boldsymbol{x}) = \log I_0 - \log I_1,$$

where the right-hand side of the equation is known constant from measurement and calibration, and dS is 1-dimensional Lebesque measure along the line

$$L(\theta, r) := \{\boldsymbol{x} \in \mathbb{R}^2 : \ x_1 \cos\theta + x_2 \sin\theta = r\}.$$

Figure 1 present parametrization of the line with it's normal direction θ and distance from origin r. For a more detailed discussion of the model describing the propagation of X-rays, see the book [37]. The entire measurement of the target $f \in L^1(\mathbb{R}^2)$ is modeled with Radon transform defined as

$$\mathcal{R}f(\theta, r) = \int_{L(\theta,r)} f(\boldsymbol{x}) dS(\boldsymbol{x}),$$

where $\theta \in [-90°, 90°)$ and $r \in \mathbb{R}$. This parametrization of tomographic data is related to so called *parallel beam geometry*, where the paths of X-rays are parallel in each direction. For visualization see figure 2. In the context of limited angle tomography, radon transform $\mathcal{R}f$ is not known on the entire angular range $\theta \in [-90°, 90°)$, but only on a subinterval $[-\phi, \phi]$ with $\phi < 90°$. Such restriction of radon transform is denoted by $\mathcal{R}_\phi f$, which is more precisely

$$\mathcal{R}_\phi f := \mathcal{R}f|_{[-\phi,\phi]\times\mathbb{R}}.$$

The inverse problem of limited angle tomography is recovering approximation of f from noisy measurements

$$m = \mathcal{R}_\phi f + \varepsilon,$$

where ε is the measurement noise.

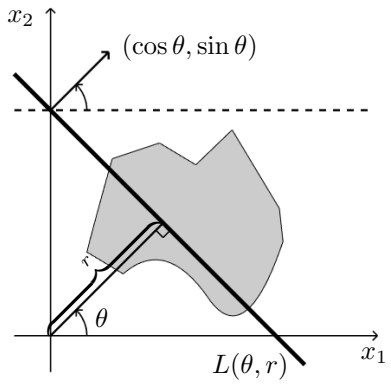


Figure 1: X-ray travels through a target that is a cartoon-like image.

The discrete setting of the X-ray tomography is required for practical applications. Discretization is done by an appropriate sampling of the continuous setting. A finite collection $\{L_j\}_{j=1}^k$ of lines $L_j \subset \mathbb{R}^2$ intersecting the domain Ω is obtained by sampling the angular variable θ and the linear parameter r uniformly over suitable intervals. Figure 2 shows an example of an (unrealistically small) collection of lines for a parallel-beam limited angle measurement, which is the case focused on the theoretical part of this thesis. Another measurement geometry called *fanbeam geometry* is visualized in figure 3. The domain is discretized by dividing it to n pixels and assuming attenuation values are constant within each pixel. The pixels are numbered from 1 to n and corresponding attenuation values are denoted by $\mathbf{f_j}$ for j in $\{1, ..., n\}$. The discretization of the domain results in the following approximation for the measurement $\mathbf{m_i}$ of the X-ray traveling through line L_i:

$$\mathbf{m_i} = \int_{L_i} f(\boldsymbol{x}) dS(\boldsymbol{x}) + \varepsilon_i \approx \sum_{j=1}^{n} a_{i,j} \mathbf{f}_j + \varepsilon_i,$$

where $a_{i,j}$ is the distance that L_i travels in the pixel indexed by j. The entire measurement process is modeled by a matrix equation $\mathbf{m} = A\mathbf{f} + \boldsymbol{\varepsilon}$, where the matrix is defined by $A = (a_{i,j})$. In some contexts discretization of the target is thought as vector in $\mathbb{R}^{N^2}$ and in others 2 dimensional object from $\mathbb{R}^{N\times N}$.

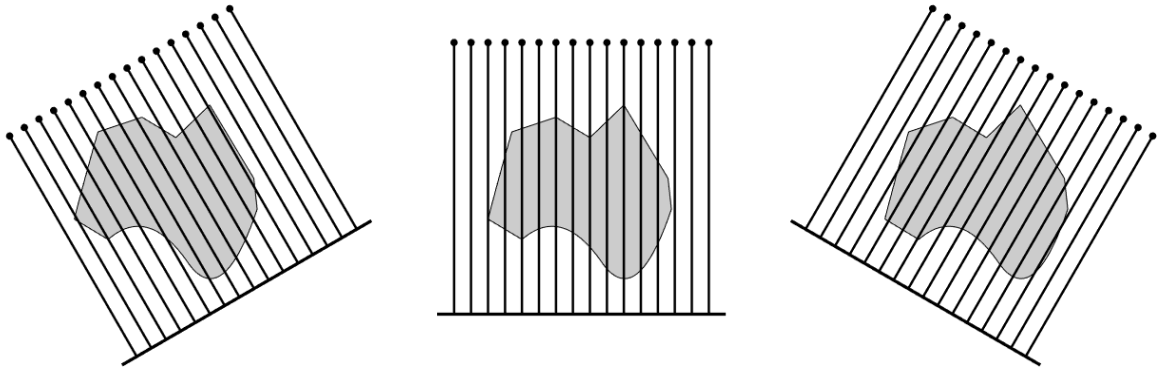


Figure 2: Visualization of parallel beam X-ray measurement geometry. Black dots represent X-ray sources, and the black line opposite side of the target represents the detector.

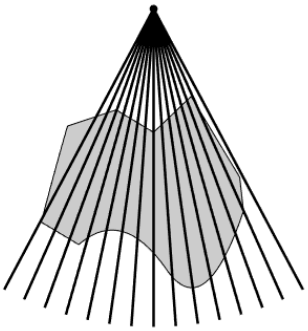


Figure 3: Visualization of fanbeam X-ray measurement geometry. For each direction there is only one common source point for all X-rays.

2.2 Fourier analysis

For a function $f \in L^1(\mathbb{R}^d)$, the *Fourier transform* of f is function from $\mathbb{R}^d$ to $\mathbb{C}$ denoted by $\mathcal{F}f(\boldsymbol{\xi})$, or $\hat{f}(\boldsymbol{\xi})$ defined as [20]

$$\widehat{f}(\boldsymbol{\xi}) = \int_{\mathbb{R}^d} f(\boldsymbol{x}) e^{-i2\pi \boldsymbol{x}\cdot\boldsymbol{\xi}} d\boldsymbol{x}.$$

The fact $f \in L^1(\mathbb{R}^d)$ guarantees that integral in the definition above is finite. The definition of Fourier transform on L^1 can also be used to define Fourier transform on L^2, since space $L^1 \cap L^2$ is dense in the space L^1 [20]. Fourier transform has the following useful properties in the space L^2.

Theorem 2. [5] *Let $f, g \in L^2(\mathbb{R}^d)$, then*

(a) $\langle f, g\rangle = \langle \hat{f}, \hat{g}\rangle$ (Parseval's relation)

(b) $\|f\|_{L^2} = \|\hat{f}\|_{L^2}$ (Plancherel's theorem)

Remark 3. In this thesis, inner product is denoted by the notation $\langle \cdot, \cdot \rangle$ except inner product of two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ is denoted by $\boldsymbol{x} \cdot \boldsymbol{y}$.

Plancerel's theorem implies that if one the of functions $f, \hat{f}$ is in the space L^2 both of them are. In fact Fourier transform is a bijection from L^2 to L^2 [20].

Some basic properties of Fourier transform are stated in the next theorem.

Theorem 4. (a) *Let $f \in L^1(\mathbb{R}^d)$*

$$\widehat{f(\cdot - \boldsymbol{y})}(\boldsymbol{\xi}) = e^{-i2\pi \boldsymbol{y}\cdot\boldsymbol{\xi}} \hat{f}(\boldsymbol{\xi})$$

(b) *Let $f \in L^1(\mathbb{R}^d)$ and $g(\boldsymbol{x}) = f(\boldsymbol{x}) e^{i2\pi \boldsymbol{x}\cdot\boldsymbol{\alpha}}$. Then*

$$\hat{g}(\boldsymbol{\xi}) = \hat{f}(\boldsymbol{\xi} - \boldsymbol{\alpha})$$

(c) *Let $A \in \mathbb{R}^{d\times d}$ be an invertible matrix and $f \in L^1(\mathbb{R}^d)$. Then*

$$\mathcal{F}(f(A\cdot))(\boldsymbol{\xi}) = |\det(A)|^{-1} \mathcal{F}f([A^{-1}]^T \boldsymbol{\xi})$$

(d) *Let $f \in L^1(\mathbb{R}^d)$ such that $\lim_{\boldsymbol{x}\to\pm\infty} f(\boldsymbol{x}) = 0$ and $\partial_j f$ exists. Then*

$$\widehat{\partial_j f}(\boldsymbol{\xi}) = i2\pi\xi_j \hat{f}(\boldsymbol{\xi})$$

Proof. Part (a) is proven first. For a function $f \in L^1(\mathbb{R}^d)$ it holds

$$\widehat{f(\cdot - \boldsymbol{y})}(\boldsymbol{\xi}) = \int_{\mathbb{R}^d} f(\boldsymbol{x} - \boldsymbol{y}) e^{-i2\pi \boldsymbol{x}\cdot\boldsymbol{\xi}} d\boldsymbol{x} = \int_{\mathbb{R}^d} f(\boldsymbol{z}) e^{-i2\pi(\boldsymbol{z}+\boldsymbol{y})\cdot\boldsymbol{\xi}} d\boldsymbol{z} = e^{-i2\pi \boldsymbol{y}\cdot\boldsymbol{\xi}} \hat{f}(\boldsymbol{\xi}).$$

The proof for (b) is quite similar and follows directly from the definition of Fourier transform and computation rule $e^a e^b = e^{a+b}$. A proof for the part (c) can be found from the book [4]. For a proof of (d) let $f \in L^1(\mathbb{R}^d)$ such that $\lim_{x\to\pm\infty} f(x) = 0$ and $\partial_j f$ exists. Using notation $\boldsymbol{x} = (x_j, \boldsymbol{x'})$ and integration by parts we get

$$\begin{aligned}
\widehat{\partial_j f}(\boldsymbol{\xi}) &= \int_{\mathbb{R}^d} \partial_j f(\boldsymbol{x}) e^{-i2\pi \boldsymbol{x}\cdot\boldsymbol{\xi}} d\boldsymbol{x} = \int_{\mathbb{R}^{d-1}} \left(\int_{\mathbb{R}} \partial_j f(x_j, \boldsymbol{x'}) e^{-i2\pi x_j \xi_j} dx_j \right) e^{-i2\pi \boldsymbol{x'}\cdot\boldsymbol{\xi'}} d\boldsymbol{x'} \\
&= \int_{\mathbb{R}^{d-1}} \Bigg(\underbrace{\lim_{R\to\infty} \Big/_{-R}^{R} f(x_j, \boldsymbol{x'}) e^{-i2\pi x_j \xi_j}}_{=0 \text{ as } \lim_{\boldsymbol{x}\to\pm\infty} f(\boldsymbol{x}) = 0} - \int_{\mathbb{R}} f(x_j, \boldsymbol{x'})(-i2\pi\xi_j) e^{-i2\pi x_j \xi_j} dx_j \Bigg) e^{-i2\pi \boldsymbol{x'}\cdot\boldsymbol{\xi'}} d\boldsymbol{x'} \\
&= i2\pi\xi_j \int_{\mathbb{R}^{d-1}} \left(\int_{\mathbb{R}} f(x_j, \boldsymbol{x'}) e^{-i2\pi x_j \xi_j} dx_j \right) e^{-i2\pi \boldsymbol{x'}\cdot\boldsymbol{\xi'}} d\boldsymbol{x'} = i2\pi\xi_j \int_{\mathbb{R}^d} f(\boldsymbol{x}) e^{-i2\pi \boldsymbol{x}\cdot\boldsymbol{\xi}} d\boldsymbol{x} \\
&= i2\pi\xi_j \hat{f}(\boldsymbol{\xi}).
\end{aligned}$$

□

If a function f is nice enough iterating the part (d) of the theorem 4 gives

$$\widehat{\partial_k^n \partial_j^m f}(\boldsymbol{\xi}) = (i2\pi)^{n+m} \xi_k^n \xi_j^m \hat{f}(\boldsymbol{\xi}).$$

This relation for the Fourier transform of partial derivatives motivates following Fourier analysis based definition for Sobolev spaces.

Definition 5. [15, 21] Let $n_1, n_2 \in \mathbb{R}_+ \cup \{0\}$, Sobolev space $H_{n_1,n_2}(\mathbb{R}^2)$ is defined as

$$H_{n_1,n_2}(\mathbb{R}^2) = \left\{ f \in L^2(\mathbb{R}^2) : \xi_1^{n_1} \xi_2^{n_2} \hat{f} \in L^2(\mathbb{R}^2) \right\}$$

Sobolev spaces has also definition based on derivatives [21]. However, the definition 5 is more general than one based on derivatives since it does not require a function to has any derivatives.

2.3 Wavefront set

Fourier analysis provides a connection between the smoothness properties of a function and the decay rate of its Fourier transform. If the function has a discontinuity, as edges in images, its Fourier transform has slow decay. The compactly supported infinitely many-times continuously differentiable functions C_0^∞ has Fourier transforms of rapid decay [20]. This motivates the following definition.

Definition 6. [31],[8] Let $f \in L^2_{loc}(\mathbb{R}^2)$, i.e. square integrable on every compact subset of $\mathbb{R}^2$. Let $\boldsymbol{x_0} \in \mathbb{R}^2$ and $s_0 \in \mathbb{R}$. We call $(\boldsymbol{x_0}, s_0)$ an *N-regular directed point of* f if there exists a smooth cut-off function $\phi \in C_0^\infty(\mathbb{R}^2)$ such that $\phi \equiv 1$ in a neighborhood $U(\boldsymbol{x_0})$ of $\boldsymbol{x_0}$ and a neighborhood $V(s_0)$ of s_0 such that there exists a constant C_N with

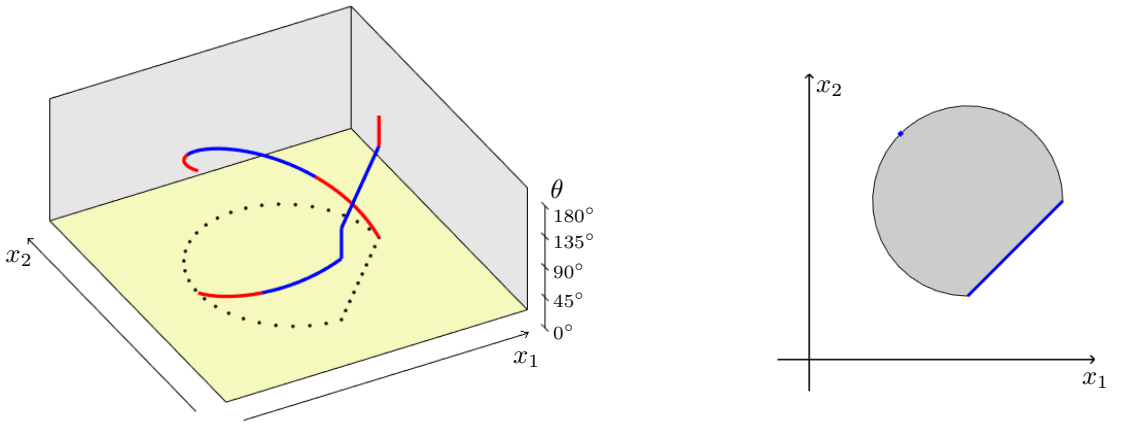
$$\left| \widehat{\phi \cdot f}(\boldsymbol{\eta}) \right| \leq C_N (1 + |\boldsymbol{\eta}|)^{-N} \qquad \text{for all } \boldsymbol{\eta} = (\eta_1, \eta_2) \text{ such that } \frac{\eta_2}{\eta_1} \in V(s_0).$$

Furthermore, we call $(\boldsymbol{x_0}, s_0)$ a *regular direct point of* f if it is N-regular directed point for every $N \in \mathbb{N}$. The *wavefront set of* f $\mathrm{WF}(f)$ is complement of the set of all regular directed points.

Remark 7. The definition of wavefront set excludes the case $s_0 = \infty$, or $\eta_1 = 0$. This problem can be avoided making the same definition with the coordinate directions reversed. This also allows to restrict attention to $s \in [-1, 1]$, since remaining directions are handled by reversing coordinate directions and considering again $s \in [-1, 1]$. For some purposes it can be useful to consider angular directions θ instead of slope-like s. These has relation $s = \tan(\theta)$.

Remark 8. The rate of decay can also be described with $\mathcal{O}$-notation. A quantity that is bounded by a constant times a when $a \to 0$ (or ∞), is denoted by $\mathcal{O}(a)$ [12].

Because the concept of wavefront set is essential in this thesis, a computational example of a wavefront set is provided here to make the definition and the concept of wavefront set clearer. Wavefront set of characteristic function of the lower half-plane $\mathbb{R}^2_- = \{(x_1, x_2) \in \mathbb{R}^2 : x_2 \leq 0\}$, f, is next proven to be $\{((0, x_2), 0) : x_2 \in \mathbb{R}\}$. Points with $x_2 > 0$ has such $\phi \in C_0^\infty$ that is supported where $f = 0$. Thus $\widehat{\phi f}(\eta) = \widehat{0}(\eta) = 0 \leq (1 + |\boldsymbol{\eta}|)^{-N}$ for every $N \in \mathbb{N}$ and $\eta \in \mathbb{R}^2$. For points with $x_2 < 0$ function ϕ can be chosen such that $\widehat{\phi f}(\boldsymbol{\eta}) = \widehat{\phi}(\boldsymbol{\eta})$. The Fourier transform $\widehat{\phi}$ is rapidly decaying as stated earlier [20]. Previous consideration showed that points with neighborhood, where f is a constant are regular directed points and doesn't belong to wavefront set. This is natural. Points, where f has a discontinuity, are considered next.



(a) (b)

Figure 4: Figure (a) provides visualization of wavefront set of indicator function f shown in figure (b). Direction dimension is presented as angles θ instead of slope-like s in definition of wavefront set to make it bounded. In figure (a) dotted line is shows projection of wavefrot set in $\mathbb{R}^2$, red line visible and blue line invisible part of wavefront set with limited angle measurement $\mathcal{R}_{45^\circ} f$. Blue points in figure (b) presents points in $\mathbb{R}^2$ which belongs to wavefront set in direction $\theta = 135^\circ$ or $s = \tan(135^\circ) = -1$.

Let choose $\phi(x_1, x_2) = \phi_1(x_1)\phi_2(x_2)$ with both $\phi_1, \phi_2 \in \mathbb{C}_0^\infty(\mathbb{R})$ such that $\operatorname{supp}(\phi_1) \in [-\delta, \delta]$. For point $(0, x_2)$ and direction $s \neq 0$, implying $\eta_2 \neq 0$, it holds

$$\begin{aligned}\left|\widehat{\phi \cdot f}(\boldsymbol{\eta})\right| &= \left|\int_{\mathbb{R}^2} f(\boldsymbol{x})\phi_1(x_1)\phi_2(x_2)e^{-i2\pi\boldsymbol{\eta}\cdot\boldsymbol{x}}d\boldsymbol{x}\right| = \left|\int_{\mathbb{R}} \phi_2(x_2)e^{-i2\pi\eta_2 x_2}dx_2 \int_{-\delta}^{0} \phi_1(x_1)e^{-i2\pi\eta_1 x_1}dx_1\right| \\ &\leq \left|\int_{\mathbb{R}} \phi_2(x_2)e^{-i2\pi\eta_2 x_2}dx_2\right| \int_{-\delta}^{0} |\phi_1(x_1)| \underbrace{|e^{-i2\pi\eta_1 x_1}|}_{\leq 1} dx_1 \leq \delta\|\phi_1\|_\infty |\hat{\phi}_2(\eta_2)|,\end{aligned}$$

where $\|\phi_1\|_\infty < \infty$ since $\phi_1 \in C_0^\infty(\mathbb{R}) \subset L^\infty(\mathbb{R})$ and Fourier transform of $\phi_2 \in C_0^\infty(\mathbb{R})$ has rapid decay, i.e.

$$\left|\widehat{\phi \cdot f}(\eta_2)\right| \leq C_N(1 + |\eta_2|)^{-N} \qquad \text{for every } N \in \mathbb{N}.$$

Let denote $\alpha = |\boldsymbol{\eta}|/|\eta_2| \geq 1$. For all η such that $\frac{\eta_2}{\eta_1} \in (s_0 - \varepsilon, s_0 + \varepsilon)$ it holds $|\frac{\eta_1}{\eta_2}| < (|s_0| - \varepsilon)^{-1}$ and therefore

$$\alpha = \frac{\sqrt{\eta_1^2 + \eta_2^2}}{|\eta_2|} \leq \frac{\sqrt{((|s_0| - \varepsilon)^{-1}\eta_2)^2 + \eta_2^2}}{|\eta_2|} = \sqrt{(|s_0| - \varepsilon)^{-1} + 1} < \infty.$$

Finally,

$$\begin{aligned}\left|\widehat{\phi \cdot f}(\boldsymbol{\eta})\right| &\leq \delta\|\phi_1\|_\infty C_N(1 + |\eta_2|)^{-N} = C'_N(1 + |\eta_2|)^{-N} = C'_N(1 + \alpha|\boldsymbol{\eta}|)^{-N} \\ &\leq C'_N(\alpha^{-1} + \alpha^{-1}|\boldsymbol{\eta}|)^{-N} = C'_N\alpha^N(1 + |\boldsymbol{\eta}|)^{-N} \qquad \text{for every } N \in \mathbb{N},\end{aligned}$$

which proves point $(0, x_2)$ with direction $s \neq 0$ doesn't belong to wavefront set of the lower half plane. Last, points $\{((0, x_2), 0) : x_2 \in \mathbb{R}\}$ are proven to belong to wavefront set of characteristic function of the lower half-plane. Using integration by parts gives

$$\begin{aligned}\widehat{\phi \cdot f}(\boldsymbol{\eta}) &= \int_{\mathrm{supp}(\phi)\cap\mathbb{R}^2_-} \phi(\boldsymbol{x}) e^{-i2\pi\eta_1 x_1} d\boldsymbol{x} = (-i2\pi\eta_1)^{-1} \int_{B(0,R)\cap\mathbb{R}^2_-} \phi(\boldsymbol{x}) \partial_1 e^{-i2\pi\eta_1 x_1} d\boldsymbol{x} \\ &= (-i2\pi\eta_1)^{-1} \left[\int_{[-R,R]\times\{0\}} \phi(\boldsymbol{x}) e^{-i2\pi\eta_1 x_1} dS(\boldsymbol{x}) - \int_{B(0,R)\cap\mathbb{R}^2_-} [\partial_1 \phi(\boldsymbol{x})] e^{-i2\pi\eta_1 x_1} d\boldsymbol{x} \right].\end{aligned}$$

It holds

$$\int_{[-R,R]\times\{0\}} \phi(\boldsymbol{x}) e^{-i2\pi\eta_1 x_1} dS(\boldsymbol{x}) = \int_{\mathbb{R}} \phi(x_1, 0) e^{-i2\pi\eta_1 x_1} dx_1 = \widehat{\phi(\cdot, 0)}(\eta_1),$$

which is nonzero for some η_1 since $\phi(x_1, 0)$ is not zero function and Fourier transform is bijective. Repeating same steps again gives

$$\widehat{\phi \cdot f}(\boldsymbol{\eta}) = \frac{\widehat{\phi(\cdot, 0)}(\eta_1)}{(-i2\pi\eta_1)} + (2\pi\eta_1)^{-2} \left[\widehat{\partial_1 \phi(\cdot, 0)}(\eta_1) - \int_{B(0,R)\cap\mathbb{R}^2_-} [\partial_1^2 \phi(\boldsymbol{x})] e^{-i2\pi\eta_1 x_1} d\boldsymbol{x} \right]$$

Because $\eta_2 = 0$ it holds

$$|\widehat{\phi \cdot f}(\boldsymbol{\eta}) - C(-i2\pi\eta_1)^{-1}| \leq (2\pi|\boldsymbol{\eta}|)^{-2} \left[2R\|\partial_1 \phi\|_\infty + \pi R^2 \|\partial_1^2 \phi\|_\infty \right],$$

which means $\widehat{\phi \cdot f}(\boldsymbol{\eta})$ is too close to $\mathcal{O}(|\boldsymbol{\eta}|^-1)$ function to satisfy

$$\left| \widehat{\phi \cdot f}(\boldsymbol{\eta}) \right| \leq C_N (1 + |\boldsymbol{\eta}|)^{-N}$$

for all $N \in \mathbb{N}$. As in this example, geometrically wavefront set contains points where function is not continuous and direction that is perpendicular to the tangent of discontinuity curve [6].

Projection of the wavefront set of f in $\mathbb{R}^2$, all points $\boldsymbol{x}$ belonging to wavefront set with some direction s, specifies singularities of function f, or edges if f is thought as an image. Edges are an essential feature of an image, and it is useful to know at what extent we can reconstruct singularities of f from CT data $\mathcal{R}_\phi f$. Wavefront set of the function f depends on properties of its Fourier transform, and the projection slice theorem relates this Fourier transform and Fourier transform of its Radon transform.

Theorem 9. *(Projection Slice Theorem) Let $f \in L^1(\mathbb{R}^2)$, $\boldsymbol{\theta}$ unit vector and $\mathcal{F}_r$ Fourier transform with respect to variable r. Then the following identity holds*

$$\mathcal{F}_r(\mathcal{R}f(\theta, r))(t) = \widehat{f}(t\boldsymbol{\theta})$$

Proof. By the definitions of the Fourier and Radon transforms it holds

$$\mathcal{F}_r(\mathcal{R}f(\theta, r))(t) = \int_{\mathbb{R}} \mathcal{R}f(\theta, r) e^{-i2\pi rt} dr = \int_{\mathbb{R}} \int_{L(\theta,r)} f(\boldsymbol{x}) dS(\boldsymbol{x}) e^{-i2\pi rt} dr.$$

In line $L(\theta, r)$ it holds $\boldsymbol{x} \cdot \boldsymbol{\theta} = x_1 \cos\theta + x_2 \sin\theta = r$, which gives

$$\int_{\mathbb{R}} \int_{L(\theta,r)} f(\boldsymbol{x}) dS(\boldsymbol{x}) e^{-i2\pi rt} dr = \int_{\mathbb{R}^2} f(\boldsymbol{x}) e^{-i2\pi \boldsymbol{x}\cdot t\boldsymbol{\theta}} d\boldsymbol{x} = \widehat{f}(t\boldsymbol{\theta}).$$

□

The projection slice theorem 9 tells measuring X-ray decay in lines $L(\theta_0, r)$ (and $\theta_0 \pm \varepsilon$) gives information only about Fourier transform of f in direction θ (and $\theta_0 \pm \varepsilon$), which is perpendicular to the lines $L(\theta_0, r)$. This suggest limited angle data $\mathcal{R}_\phi f$ doesn't contain information to reconstruct singularities in directions not contained in interval $[-\phi, \phi]$. The precise connection between the wavefront sets of f and $\mathcal{R}_\phi f$ and a proof for it is provided in article [39]:

Theorem 10. *Let $f \in L^2_{loc}(\mathbb{R}^2)$ and $L_0 = L(\theta_0, r_0)$ be a line in the plane. Let $(\boldsymbol{x_0}, s) \in \mathrm{WF}(f)$ such that $\boldsymbol{x_0} \in L_0$ and $s = \tan(\theta_0)$. Then it holds:*

(i) *The singularity of f at $(\boldsymbol{x_0}, s)$ cause a unique singularity in $\mathrm{WF}(\mathcal{R}f)$ at (θ_0, r_0)*

(ii) *Singularities of f not tangent to $L(\theta_0, r_0)$ do not cause singularities in $\mathcal{R}f$ at (θ_0, r_0).*

Singularities that can be reconstructed from limited angle data $\mathcal{R}_\phi f$ are called *visible* and those that can not *invisible* part of wavefront set.

2.4 Frames

This subsection is based on books [12, 31]. The idea of frames is to provide more general building blocks, or representation system, than bases for elements of a Hilbert space. This system should still have some nice properties: every element should be able to represent as a superposition of these building blocks and frame coefficients, inner products of function with building blocks should characterize function. Moreover, the function should be able to reconstruct stable from these coefficients. What is lost while gaining more generality is the uniqueness of representations. The definition of frame is provided next and how it is related niceness of the representation system is discussed after that.

Definition 11. Sequence $(\varphi_j)_{j\in J}$ in a Hilbert space $\mathcal{H}$ is *frame for $\mathcal{H}$*, if there exists constants $0 < A$ and $B < \infty$ such that

$$A\|f\|^2 \leq \sum_{j\in J} |\langle f, \varphi_j\rangle|^2 \leq B\|f\|^2 \quad \text{for all } f \in \mathcal{H}.$$

A frame is called *tight* if inequality above holds for $A = B$.

Let's focus first on why frame coefficients $(\langle f, \varphi_j\rangle)_{j\in J}$ characterize function f. Norms $\|\cdot\| = \|\cdot\|_{\mathcal{H}}$ and $\|\cdot\|_{\ell^2} = \sqrt{\sum_{j\in J} |\cdot|^2}$ define metrics, way to measure distances, for Hilbert space $\mathcal{H}$ and coefficient space ℓ^2. If $f, g \in \mathcal{H}$, replacing f in definition on frames with $f - g$ implies closeness in both spaces are related. If coefficients $(\langle f, \varphi_j\rangle)_{j\in J}$ are close to $(\langle g, \varphi_j\rangle)_{j\in J}$, then by the linearity of inner product and left inequality of definition 11 also f is close to g and vise versa by right inequality of definition. Exact characterization follows from this, since frame coefficients are equal, their distance is zero, if and only if functions are equal.

To discuss reconstruction and superposition properties of frame some frame theory is provided first. Function

$$T : \mathcal{H} \to \ell^2(J), \quad f \mapsto (\langle f, \varphi_j\rangle)_{j\in J}$$

is called the *analysis operator* since frame coefficents $(\langle f, \varphi_j\rangle)_{j\in J}$ provide way to analyze function f. The adjoint of analysis operator

$$T^* : \ell^2(J) \to \mathcal{H}, \quad ((c)_{j\in J}) \mapsto \sum_{j\in J} c_j \varphi_j,$$

is referred to as *synthesis operator*. The main operator accosiated with a frame is *frame operator*

$$S = T^*T : \mathcal{H} \to \mathcal{H}, \quad f \mapsto \sum_{j\in J} \langle f, \varphi_j\rangle \varphi_j,$$

which can be proven to be invertible and self-adjoint operator. Sequence $(\tilde{\varphi}_j)_{j\in J} = (S^{-1}\varphi_j)_{j\in J}$ also forms frame that is called *canonical dual frame*. By linearity of S^{-1} we have

$$f = S^{-1}Sf = S^{-1}(\sum_{j\in J}\langle f,\varphi_j\rangle\varphi_j) = \sum_{j\in J}\langle f,\varphi_j\rangle S^{-1}\varphi_j = \sum_{j\in J}\langle f,\varphi_j\rangle\tilde{\varphi}_j \tag{12}$$

and using

$$\langle f,(S^{-1})^*\varphi_j\rangle = \langle S^{-1}f,\varphi_j\rangle = \langle S^{-1}f,SS^{-1}\varphi_j\rangle = \langle S^{-1}f,S^*S^{-1}\varphi_j\rangle = \langle SS^{-1}f,S^{-1}\varphi_j\rangle = \langle f,S^{-1}\varphi_j\rangle$$

gives

$$f = SS^{-1}f = T^*(T(S^{-1}f)) = T^*((\langle S^{-1}f,\varphi_j\rangle)_{j\in J}) = T^*((\langle f,S^{-1}\varphi_j\rangle)_{j\in J}) = \sum_{j\in J}\langle f,\tilde{\varphi}_j\rangle\varphi_j. \tag{13}$$

Therefore by equations (12) and (13) reconstruction of function from frame coefficients and presenting function as superposition of frame elements requires only finding canonical dual frame and algorithm for it is known.

2.5 Shearlets

Fourier analysis provides a classic example of representing and analyzing functions using a collection of functions. In this case, these functions, trigonometric or complex exponentials, forms a basis. Despite the great power of Fourier analysis, it has a drawback. The basis functions of Fourier analysis are similar over whole space, and thus the representation of data with them is not capable of localizing properties of the analyzed function. It is possible to tell that a function has a point of discontinuity from the decay rate of its Fourier transform, but not where it is. Functions known as wavelets overcome this drawback and are localized in space. When considering the higher dimension basic concept of wavelets is not that optimal anymore, since they are *isotropic*. It means they treat every direction in space equally. The need for anisotropic representation system has lead to the rise of shearlets. Shearlets do not constitute a basis but frame, which still guarantees them to have some nice properties.

Shearlets are based on changing resolution, orientation and position of input of certain generating functions. Typical choices of these generating functions are either compactly supported or band limited, i.e. compactly supported on frequency domain. In 2D case resolution of input is changed by multiplication with a parabolic scaling matrix

$$A_a = \begin{bmatrix} a & 0 \\ 0 & a^{1/2} \end{bmatrix} \quad \text{or} \quad \tilde{A}_a = \begin{bmatrix} a^{1/2} & 0 \\ 0 & a \end{bmatrix},$$

orientation of input by multiplication with a shearing matrix $S_s = \begin{bmatrix} 1 & -s \\ 0 & 1 \end{bmatrix}$ or S_s^T and position of input by translation. There are also other ways to change the orientation of a function, like the rotation matrix used by curvelets. The use of the shearíng matrix is preferred as it preserves integer lattice if s is an integer. This is a nice property for practical implementations. The next definition describes the property, which the used generating functions are wanted to fulfill.

Definition 14. [21, 31] A function $\psi \in L^2(\mathbb{R}^2)$ is *admissible* if

$$\int_{\mathbb{R}^2} \frac{|\hat{\psi}(\boldsymbol{\xi})|^2}{|\xi_1|^2} d\boldsymbol{\xi} < \infty.$$

Admissible functions are called *shearlets*.

Considerations above motivate *continuous shearlet system*

$$\left\{\psi_{a,s,\boldsymbol{t}} = a^{-\frac{3}{4}}\psi(A_a^{-1}S_s^{-1}(\cdot - \boldsymbol{t})) : a \in \mathbb{R}\setminus\{0\}, s \in \mathbb{R}, \boldsymbol{t} \in \mathbb{R}^2\right\} \tag{15}$$

Instead of the system above cone-adapted shearlet system is used in practice, since it allows to limit values of s to a finite interval. This avoids directional bias following from the fact that directions indexed by s are not equally distributed as s grows large.

Definition 16. [31] For functions $\phi, \psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$, the *cone-adapted continuous shearlet system* $SH(\phi, \psi, \tilde{\psi})$ is defined by

$$SH(\phi, \psi, \tilde{\psi}) = \Phi(\phi) \cup \Psi(\psi) \cup \tilde{\Psi}(\tilde{\psi}),$$

where

$$\begin{aligned}
\Phi(\phi) &= \left\{\phi_{\boldsymbol{t}} = \phi(\cdot - \boldsymbol{t}) : \boldsymbol{t} \in \mathbb{R}^2\right\},\\
\Psi(\psi) &= \left\{\psi_{a,s,\boldsymbol{t}} = a^{-\frac{3}{4}}\psi(A_a^{-1}S_s^{-1}(\cdot - \boldsymbol{t})) : a \in (0,1], |s| \le 1 + a^{1/2}, \boldsymbol{t} \in \mathbb{R}^2\right\},\\
\tilde{\Psi}(\tilde{\psi}) &= \left\{\tilde{\psi}_{a,s,\boldsymbol{t}} = a^{-\frac{3}{4}}\psi(\tilde{A}_a^{-1}S_s^{-1}(\cdot - \boldsymbol{t})) : a \in (0,1], |s| \le 1 + a^{1/2}, \boldsymbol{t} \in \mathbb{R}^2\right\}.
\end{aligned}$$

Term cone-adapted refers to fact that it is adapted to cone like partioning of frequence plane. It is partitioned to low-frequency square Φ, horizontal and vertical conic regions Ψ and $\tilde{\Psi}$, see figure 5. Lets denote $\mathbb{S}_{\text{cone}} = \{(a, s, \boldsymbol{t}) : a \in (0,1], |s| \le 1 + a^{1/2}, \boldsymbol{t} \in \mathbb{R}^2\}$. *Shearlet transform* of $f \in L^2(\mathbb{R}^2)$ assosiated with shearlet system $SH(\phi, \psi, \tilde{\psi})$ is mapping defined by

$$f \to \mathcal{SH}_{\phi,\psi,\tilde{\psi}} f(\boldsymbol{t}', (a, s, \boldsymbol{t}), (\tilde{a}, \tilde{s}, \tilde{\boldsymbol{t}})) = (\langle f, \phi_{\boldsymbol{t}'}\rangle, \langle f, \psi_{a,s,\boldsymbol{t}}\rangle, \langle f, \tilde{\psi}_{\tilde{a},\tilde{s},\tilde{\boldsymbol{t}}}\rangle),$$

where

$$(\boldsymbol{t}', (a, s, \boldsymbol{t}), (\tilde{a}, \tilde{s}, \tilde{\boldsymbol{t}})) \in \mathbb{R}^2 \times \mathbb{S}^2_{\text{cone}}.$$

Continuous shearlets are useful for theoretical purposes, but discrete setting is required for practical applications. Discrete shearlet systems are obtained by sampling the continuous ones. A discrete shearlet system constitutes frame with appropriate construction of shearlet generating function and sample of scale, shearing and translation parameters [31].

Definition 17. [31] For functions $\phi, \psi, \tilde{\psi} \in L^2(\mathbb{R}^2)$ and parameter $c = (c_1, c_2) \in (\mathbb{R}_+)^2$, the (*regular*) *cone-adapted discrete shearlet system* $SH(\phi, \psi, \tilde{\psi}; c)$ is defined by

$$SH(\phi, \psi, \tilde{\psi}; c) = \Phi(\phi; c_1) \cup \Psi(\psi; c) \cup \tilde{\Psi}(\tilde{\psi}; c),$$

where

$$\begin{aligned}
\Phi(\phi; c_1) &= \left\{\phi_m = \phi(\cdot - c_1 m) : m \in \mathbb{Z}^2\right\},\\
\Psi(\psi; c) &= \left\{\psi_{j,k,m} = 2^{\frac{3}{4}j}\psi(S_k A_{2^j} \cdot - M_c m) : j \ge 0, |k| \le \lceil 2^{j/2}\rceil, m \in \mathbb{Z}^2\right\},\\
\tilde{\Psi}(\tilde{\psi}; c) &= \left\{\psi_{j,k,m} = 2^{\frac{3}{4}j}\tilde{\psi}(S_k^T \tilde{A}_{2^j} \cdot - \tilde{M}_c m) : j \ge 0, |k| \le \lceil 2^{j/2}\rceil, m \in \mathbb{Z}^2\right\},
\end{aligned}$$

with

$$M_c = \begin{bmatrix} c_1 & 0 \\ 0 & c_2 \end{bmatrix} \quad \text{and} \quad \tilde{M}_c = \begin{bmatrix} c_2 & 0 \\ 0 & c_1 \end{bmatrix}.$$

Denote $\Lambda = \mathbb{N}_0 \times \{-\lceil 2^{j/2}\rceil, ..., \lceil 2^{j/2}\rceil\} \times \mathbb{Z}^2$. *The cone-adapted discrete shearlet transform* of $f \in L^2(\mathbb{R}^2)$ assosiated with shearlet system $SH(\phi, \psi, \tilde{\psi}; c)$ is mapping defined by

$$f \to \mathcal{SH}_{\phi,\psi,\tilde{\psi}} f(m', (j, k, m), (\tilde{j}, \tilde{k}, \tilde{m})) = (\langle f, \phi_{m'}\rangle, \langle f, \psi_{j,k,m}\rangle, \langle f, \tilde{\psi}_{\tilde{j},\tilde{k},\tilde{m}}\rangle),$$

where

$$(m', (j, k, m), (\tilde{j}, \tilde{k}, \tilde{m})) \in \mathbb{Z}^2 \times \Lambda \times \Lambda.$$

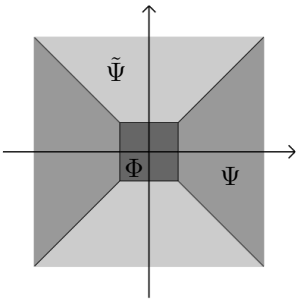


Figure 5: Partition of the frequency plane

2.6 Shearlets and wavefront set

Wavefront set of a function f is characterized by the decay rate of its shearlet coefficients. Shearlet coefficients in point $\boldsymbol{t_0}$ and direction s_0 decay slow if and only if $(\boldsymbol{t_0}, s_0)$ belongs to wavefront set of f. This subsection gives proof of one direction of this result. If a point does not belong to the wavefront set, then shearlet transform in that point has "fast" decay as a, scale parameter of the shearlet system, goes to zero. Another direction, fast shearlet decay implies point does not belong to the wavefront set, is proven in the article [21].

Definition 18. [21] Function ψ is said to have *n-vanishing moments in x_1-direction* if

$$\int_{\mathbb{R}^2} \frac{|\hat{\psi}(\boldsymbol{\xi})|^2}{|\xi_1|^{2n}} d\boldsymbol{\xi} < \infty.$$

Term n-vanishing moments is motivated by the fact that this definition is almost equal to condition $\int_{\mathbb{R}} x_1^k \psi(x_1, x_2) dx_1 = 0$ for all $x_2 \in \mathbb{R}, k < n$. The requirement that shearlet has n-vanishing moments in x_1-direction is practical to fulfill and allows constructions of shearlet frames with compact support [21].

Lemma 19. *Let ψ be shearlet with n-vanishing moments in x_1-direction. Then there exists $\theta \in L^2(\mathbb{R}^2)$ such that*

$$\hat{\psi}(\xi_1, \xi_2) = \xi_1^n \hat{\theta}(\xi_1, \xi_2), \tag{20}$$

for $\xi_1 \in \mathbb{R} \setminus \{0\}, \xi_2 \in \mathbb{R}$.

Proof. Shearlet ψ is L^2-function and thus Fourier transform $\hat{\psi}$ exists. For $\boldsymbol{\xi} \in \Gamma = \{\xi_1 \in \mathbb{R} \setminus \{0\}, \xi_2 \in \mathbb{R}\}$ define

$$g(\boldsymbol{\xi}) = \frac{\hat{\psi}(\boldsymbol{\xi})}{\xi_1^n}.$$

Now proof $g \in L^2(\mathbb{R}^2)$:

$$\|g\|_2^2 = \int_{\mathbb{R}^2} g(\boldsymbol{\xi})^2 d\boldsymbol{\xi} = \int_{\Gamma} g(\boldsymbol{\xi})^2 d\boldsymbol{\xi} = \int_{\Gamma} \frac{|\hat{\psi}(\boldsymbol{\xi})|^2}{|\xi_1|^{2n}} d\boldsymbol{\xi} = \int_{\mathbb{R}^2} \frac{|\hat{\psi}(\boldsymbol{\xi})|^2}{|\xi_1|^{2n}} d\boldsymbol{\xi},$$

which is finite as ψ has n varishing moments in x_1-direction. Every choice for values of g for set $\mathbb{R}^2 \setminus \Gamma$ keeps g as L^2-function because set of measure zero has no affect on value of integral. As Fourier transform bijection from L^2 to L^2 there exists θ such that $\hat{\theta} = g$. □

Theorem 21. [21](Direct Theorem) *Assume that $f \in L^2(\mathbb{R}^2)$ and $(\boldsymbol{t_0}, s_0)$ is an N-regular directed point of f. Let $\psi \in H_{0,L}(\mathbb{R}^2), \hat{\psi} \in L^1(\mathbb{R}^2)$ be a shearlet with M moments which satisfies a decay estimate of the form*

$$\psi(\boldsymbol{x}) \leq C(1+|\boldsymbol{x}|)^{-P}. \tag{22}$$

Then there exists a neighborhood $U(\boldsymbol{t_0})$ of $\boldsymbol{t_0}$ and $V(s_0)$ of s_0 such that for any $1/2 < \alpha < 1, \boldsymbol{t} \in U(\boldsymbol{t_0})$ and $s \in V(s_0)$ shearlet coefficients $\mathcal{SH}_\psi f(a,s,\boldsymbol{t})$ has following decay rate as $a \to 0$

$$\mathcal{SH}_\psi f(a,s,\boldsymbol{t}) \in \mathcal{O}(a^{-3/4+P/2} + a^{(1-\alpha)M} + a^{-3/4+\alpha N} + a^{(\alpha-1/2)L}) \tag{23}$$

Proof. First we show following decay rate for difference of f and ϕf

$$|\langle (1-\phi)f, \psi_{a,s,\boldsymbol{t}} \rangle| \in \mathcal{O}(a^{-3/4+P/2}),$$

where $\phi \in C_0^\infty(\mathbb{R}^2)$ is as in definition of wavefront set 6. By (22) we estimate

$$\begin{aligned}
|\psi_{a,s,\boldsymbol{t}}(\boldsymbol{x})| &\leq Ca^{-3/4}\big(1+|\begin{bmatrix} a^{-1} & sa^{-1} \\ 0 & a^{-1/2} \end{bmatrix}(\boldsymbol{x}-\boldsymbol{t})|\big)^{-P} \\
&\leq Ca^{-3/4}\big(1+|\begin{bmatrix} 0 & 0 \\ 0 & a^{-1/2} \end{bmatrix}(\boldsymbol{x}-\boldsymbol{t})|\big)^{-P} \\
&= Ca^{-3/4}(1+a^{-1/2}|x_2-t_2|)^{-P} \in \mathcal{O}(a^{-3/4+P/2}|x_2-t_2|^{-P}).
\end{aligned}$$

Definition of wavefront set implies $(1-\phi)f = 0$ in a small neighborhood $U(\boldsymbol{t_0})$ of $\boldsymbol{t_0}$. Therefore we get

$$\begin{aligned}
|\langle (1-\phi)f, \psi_{a,s,\boldsymbol{t}} \rangle| &\leq \int_{\mathbb{R}^2} |(1-\phi(\boldsymbol{x}))f(\boldsymbol{x})||\psi_{a,s,\boldsymbol{t}}(\boldsymbol{x})|d\boldsymbol{x} \\
&\leq Ca^{-3/4+P/2} \int_{|x_2-t_2|\geq\delta} |x_2-t_2|^{-P}|1-\phi(\boldsymbol{x})||f(\boldsymbol{x})|d\boldsymbol{x} \\
&\leq Ca^{-3/4+P/2}\delta^{-P}(1+\|\phi\|_\infty)\|f\|_2 \in \mathcal{O}(a^{-3/4+P/2}),
\end{aligned} \tag{24}$$

since $\phi \in C_0^\infty(\mathbb{R}^2) \subset L^\infty(\mathbb{R}^2)$ and $f \in L^2(\mathbb{R}^2)$.

Remaining part of proof shows sufficient decay estimate for $|\langle \phi f, \psi_{a,s,\boldsymbol{t}} \rangle|$. It will use the fact that the shearlet

$$\psi_{a,s,\boldsymbol{t}}(\boldsymbol{x}) = a^{-3/4}\psi(A_a^{-1}S_s^{-1}(\boldsymbol{x}-\boldsymbol{t})) = a^{-3/4}\psi(\frac{(x_1-t_1)+s(x_2-t_2)}{a}, \frac{x_2-t_2}{a^{1/2}})$$

and thus by basic properties of Fourier transform

$$\hat{\psi}_{a,s,\boldsymbol{t}}(\boldsymbol{\xi}) = a^{3/4}e^{-i2\pi\boldsymbol{t}\cdot\boldsymbol{\xi}}\hat{\psi}(a\xi_1, a^{1/2}(\xi_2 - s\xi_1)).$$

For $\alpha \in (\frac{1}{2}, 1)$ we can write

$$\begin{aligned}
|\langle \phi f, \psi_{a,s,\boldsymbol{t}} \rangle| &\overset{2(a)}{=} |\langle \widehat{\phi f}, \hat{\psi}_{a,s,\boldsymbol{t}} \rangle| \leq a^{3/4} \int_{\mathbb{R}^2} |\widehat{\phi f}(\boldsymbol{\xi})||\hat{\psi}_{a,s,\boldsymbol{t}}(\boldsymbol{\xi})|d\boldsymbol{\xi} \\
&= \underbrace{\int_{|\xi_1|<a^{-\alpha}} |\widehat{\phi f}(\boldsymbol{\xi})||\hat{\psi}_{a,s,\boldsymbol{t}}(\boldsymbol{\xi})|d\boldsymbol{\xi}}_{A} + \underbrace{\int_{|\xi_1|>a^{-\alpha}} |\widehat{\phi f}(\boldsymbol{\xi})||\hat{\psi}_{a,s,\boldsymbol{t}}(\boldsymbol{\xi})|d\boldsymbol{\xi}}_{B}
\end{aligned}$$

Using lemma 19 and properties of Fourier transform gives

$$A = \int_{|\xi_1|<a^{-\alpha}} |\widehat{\phi f}(\boldsymbol{\xi})||a^{3/4}e^{-i2\pi \boldsymbol{t}\cdot\boldsymbol{\xi}}\hat{\psi}(a\xi_1, a^{1/2}(\xi_2 - s\xi_1))|d\boldsymbol{\xi} \tag{25}$$

$$= \int_{|\xi_1|<a^{-\alpha}} a^M|\xi_1|^M|\widehat{\phi f}(\boldsymbol{\xi})||\underbrace{a^{3/4}e^{-i2\pi \boldsymbol{t}\cdot\boldsymbol{\xi}}\hat{\theta}(a\xi_1, a^{1/2}(\xi_2 - s\xi_1))}_{=:\ \hat{\theta}_{a,s,\boldsymbol{t}}}|d\boldsymbol{\xi}$$

$$\leq a^{M(1-\alpha)}\int_{|\xi_1|<a^{-\alpha}} |\widehat{\phi f}(\boldsymbol{\xi})||\hat{\theta}_{a,s,\boldsymbol{t}}(\boldsymbol{\xi})|d\boldsymbol{\xi}$$

$$\leq a^{M(1-\alpha)}\langle|\widehat{\phi f}|, |\hat{\theta}_{a,s,\boldsymbol{t}}|\rangle \overset{\text{Schwarz}}{\leq} a^{M(1-\alpha)}\|\widehat{\phi f}\|_2\|\hat{\theta}_{a,s,\boldsymbol{t}}\|_2, \tag{26}$$

and $\|\widehat{\phi f}\|_2\|\hat{\theta}_{a,s,\boldsymbol{t}}\|_2$ is finite by Plancherel's theorem, since both, ϕf and $\theta_{a,s,\boldsymbol{t}}$, are L^2-functions.

To estimate B we make the following substitution

$$([A_a^{-1}S_s^{-1}]^{-1})^T\boldsymbol{\xi} = \begin{bmatrix} a & 0 \\ -sa^{1/2} & a^{1/2} \end{bmatrix}\boldsymbol{\xi} = \tilde{\boldsymbol{\xi}}, \qquad d\tilde{\boldsymbol{\xi}} = |\det\left(([A_a^{-1}S_s^{-1}]^{-1})^T\right)|d\boldsymbol{\xi} = a^{3/2}d\boldsymbol{\xi}.$$

Then

$$B = a^{-3/4}\int_{\frac{|\tilde{\xi}_1|}{a}>a^{-\alpha}} |\widehat{\phi f}(\frac{\tilde{\xi}_1}{a}, \frac{s}{a}\tilde{\xi}_1 + a^{-1/2}\tilde{\xi}_2)||\hat{\psi}(\tilde{\boldsymbol{\xi}})|d\tilde{\boldsymbol{\xi}}. \tag{27}$$

Now we use the fact that $(\boldsymbol{t_0}, s_0)$ is an N-regular directed point of f. This means there is a neighborhood $(s_0 - \varepsilon, s_0 + \varepsilon)$ such that

$$|\widehat{\phi f}(\boldsymbol{\eta})| \leq C(1 + |\boldsymbol{\eta}|)^{-N} \qquad \text{for all } \boldsymbol{\eta} \text{ such that } \frac{\eta_2}{\eta_1} \in (s_0 - \varepsilon, s_0 + \varepsilon). \tag{28}$$

Looking at (27) we now consider $\frac{\eta_2}{\eta_1}$ with $\eta_1 := \frac{\tilde{\xi}_1}{a}$, $\eta_2 := \frac{s}{a}\tilde{\xi}_1 + a^{-1/2}\tilde{\xi}_2$ and $\frac{|\tilde{\xi}_1|}{a} > a^{-\alpha}$ and get the estimate

$$s - a^{\alpha-1/2}|\tilde{\xi}_2| < s - a^{-1/2}|\tilde{\xi}_2|\frac{a}{|\tilde{\xi}_1|} \leq s + a^{-1/2}\tilde{\xi}_2\frac{a}{\tilde{\xi}_1} = \frac{\eta_2}{\eta_1} \leq s + a^{-1/2}|\tilde{\xi}_2|\frac{a}{|\tilde{\xi}_1|} < s + a^{\alpha-1/2}|\tilde{\xi}_2| \tag{29}$$

By (28) we have that

$$|\widehat{\phi f}(\frac{\tilde{\xi}_1}{a}, \frac{s}{a}\tilde{\xi}_1 + a^{-1/2}\tilde{\xi}_2)| \leq C\big(1 + \Big|(\frac{\tilde{\xi}_1}{a}, \frac{s}{a}\tilde{\xi}_1 + a^{-1/2}\tilde{\xi}_2)\Big|\big)^{-N} \leq C\big(1 + \frac{|\tilde{\xi}_1|}{a}\big)^{-N} \tag{30}$$

for s in a neighborhood $V(s_0)$ of s_0, $\frac{|\tilde{\xi}_1|}{a} > a^{-\alpha}$ and $|\tilde{\xi}_2| < \varepsilon' a^{1/2-\alpha}$ for some $\varepsilon' < \varepsilon$. Now integral B is split according to

$$B = a^{-3/4}\int_{\frac{|\tilde{\xi}_1|}{a}>a^{-\alpha}} |\widehat{\phi f}(\frac{\tilde{\xi}_1}{a}, \frac{s}{a}\tilde{\xi}_1 + a^{-1/2}\tilde{\xi}_2)||\hat{\psi}(\tilde{\boldsymbol{\xi}})|d\tilde{\boldsymbol{\xi}}$$

$$= \underbrace{a^{-3/4}\int_{\frac{|\tilde{\xi}_1|}{a}>a^{-\alpha}, |\tilde{\xi}_2|<\varepsilon' a^{1/2-\alpha}}}_{B_1} + \underbrace{a^{-3/4}\int_{\frac{|\tilde{\xi}_1|}{a}>a^{-\alpha}, |\tilde{\xi}_2|>\varepsilon' a^{1/2-\alpha}}}_{B_2}.$$

By (30) we can estimate.

$$B_1 \leq Ca^{\alpha N - 3/4}\|\hat{\psi}\|_1 \tag{31}$$

It remains to estimate B_2, which can be done using assumption $\psi \in H_{0,L}(\mathbb{R}^2)$:

$$\begin{aligned}
B_2 &= a^{-3/4} \int_{\frac{|\tilde{\xi}_1|}{a} > a^{-\alpha}, |\tilde{\xi}_2| > \varepsilon' a^{1/2-\alpha}} |\widehat{\phi f}(\frac{\tilde{\xi}_1}{a}, \frac{s}{a}\tilde{\xi}_1 + a^{-1/2}\tilde{\xi}_2)||\hat{\psi}(\tilde{\boldsymbol{\xi}})| d\tilde{\boldsymbol{\xi}} \\
&= a^{-3/4} \int_{\frac{|\tilde{\xi}_1|}{a} > a^{-\alpha}, |\tilde{\xi}_2| > \varepsilon' a^{1/2-\alpha}} |\widehat{\phi f}(\frac{\tilde{\xi}_1}{a}, \frac{s}{a}\tilde{\xi}_1 + a^{-1/2}\tilde{\xi}_2)||\xi_2^{-L}\xi_2^L \hat{\psi}(\tilde{\boldsymbol{\xi}})| d\tilde{\boldsymbol{\xi}} \\
&\leq (\varepsilon')^{-L} a^{-3/4+(\alpha-1/2)L} \int_{\frac{|\tilde{\xi}_1|}{a} > a^{-\alpha}, |\tilde{\xi}_2| > \varepsilon' a^{1/2-\alpha}} |\widehat{\phi f}(\frac{\tilde{\xi}_1}{a}, \frac{s}{a}\tilde{\xi}_1 + a^{-1/2}\tilde{\xi}_2)||\xi_2^L \hat{\psi}(\tilde{\boldsymbol{\xi}})| d\tilde{\boldsymbol{\xi}} \\
&= (\varepsilon')^{-L} a^{(\alpha-1/2)L} \langle |\widehat{\phi f}|, |\xi_2^L \hat{\psi}| \rangle \overset{\text{Schwarz}}{\leq} a^{(\alpha-1/2)L} \|\widehat{\phi f}\|_2 \|\xi_2^L \hat{\psi}\|_2, \qquad (32)
\end{aligned}$$

where $\|\widehat{\phi f}\|_2$ is finite by Plancherel's theorem, since ϕf is L^2-function and $\|\xi_2^L \hat{\psi}\|_2$ finite as $\psi \in H_{0,L}(\mathbb{R}^2)$.

Combining estimates (24), (25), (31), (32) gives

$$|\langle f, \psi_{a,s,t} \rangle| \leq (|\langle (1-\phi)f, \psi_{a,s,t} \rangle| + |\langle \phi f, \psi_{a,s,t} \rangle|) \in \mathcal{O}(a^{-3/4+P/2} + a^{(1-\alpha)M} + a^{-3/4+\alpha N} + a^{(\alpha-1/2)L}).$$

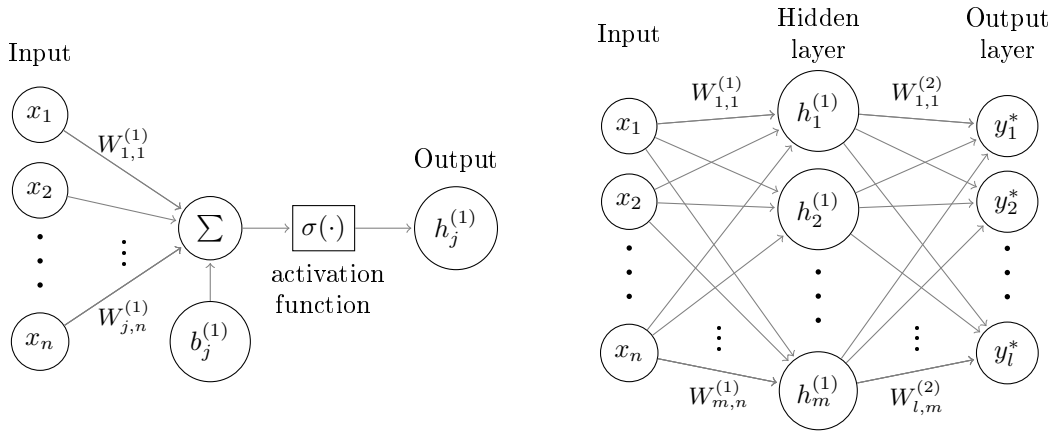
□

3 Neural Networks

This section is based on the book [19], unless otherwise cited. Feedforward neural network, the quintessential deep learning model, is a function of the following form, $\mathcal{NN}_\theta(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^m$,

$$\begin{aligned}\mathcal{NN}_\theta(\boldsymbol{x}) &= \sigma_L(W^{(L)}\sigma_{L-1}(W^{(L-1)}\sigma_{L-2}(...(\underbrace{\sigma_1(W^{(1)}\boldsymbol{x}+\boldsymbol{b}^{(1)})}_{=\boldsymbol{h}^{(1)}})...)+\boldsymbol{b}^{(L-1)})+\boldsymbol{b}^{(L)}) \\ &= \sigma_L(W^{(L)}\boldsymbol{h}^{(L-1)}+\boldsymbol{b}^{(L)}),\end{aligned} \tag{33}$$

where $L \in \mathbb{N}$ is the *depth* of the model and for each $k \in \{1, ..., L\}$, $W^{(k)}$ is a linear function (*weights*), $\boldsymbol{b}^{(k)}$ a vector (*biases*) and $\sigma_k : \mathbb{R} \to \mathbb{R}$ a (non-linear) activation function applied element-wise. Note that a finite input of any shape can be reshaped to a vector $\boldsymbol{x} \in \mathbb{R}^n$, which makes the formalization above more general. The structure $\sigma_k(W^{(k)}(\cdot))$ is called *k:th layer* of the network. The name "deep learning" is motivated by the fact that practical neural networks have typically multiple layers. Other layers than the output layer are called hidden layers because training data does not specify what the output from these layers should be. Instead, the learning algorithm decides how to set them to produce the desired output from the network. The letter θ in $\mathcal{NN}_\theta$ refers to all parameters of the neural network, weights and biases, which are learned to approximate the desired relation. Learning the parameters is typically minimization of a cost function using a gradient-based algorithm. For visualization of feedforward neural network, see figure 6 and for examples of activation functions figure 7. Note that in theory, convolutional neural networks (CNNs) are a special case of feedforward neural networks. This is true since convolution is a linear operator and thus can be presented as matrix multiplication. CNNs are presented in section 3.2.



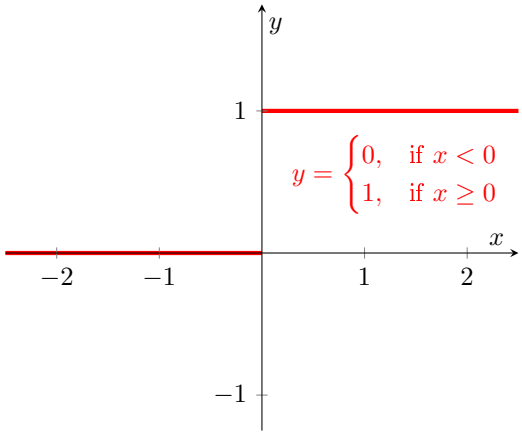
(a) An artificial neuron

(b) A neural network with one hidden layer

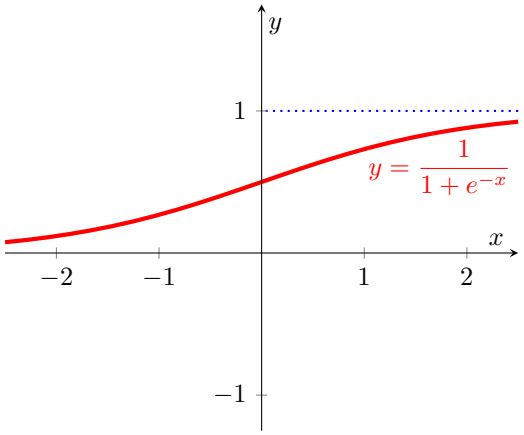
Figure 6: Visualization of a neural network. In figure (b) each node of hidden and output layers represents the output of a neuron computed from the preceding layer as presented in figure (a). Neural networks like in figure (b) are called *fully connected*. This is motivated by the fact that for all layers each input node has a connection to each output node.

Neural networks are called *neural* because they are loosely inspired by neuroscience. Instead of vector-valued functions, layers of the neural network can be interpreted as multiple parallel vector-to-scalar function units, artificial neurons. They are similar to biological neurons since both have multiple inputs, from the outputs of other neurons, and compute its activation value, see figure 6a. Activation of biological

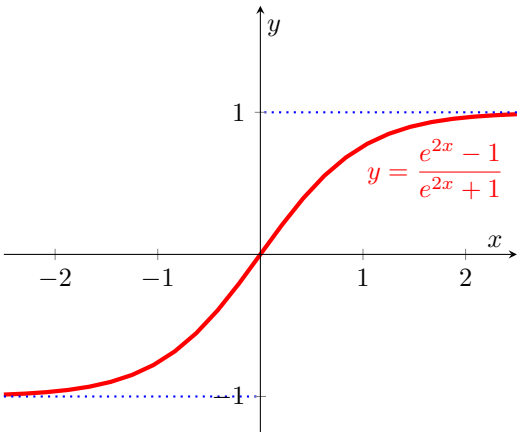
neurons is modeled with a step function, but parameters of a neural network with some other activation functions are easier to optimize. Some practical activation functions motivated by step function are presented in figure 7. Sigmoid and hyperbolic tangent are smoothed versions of step function onto intervals $(0, 1)$ and $(-1, 1)$. These step-like functions have derivative close to zero, or zero across most of their domain, which can make optimization hard. The rectified linear unit has a better behaving derivative when the unit is active.



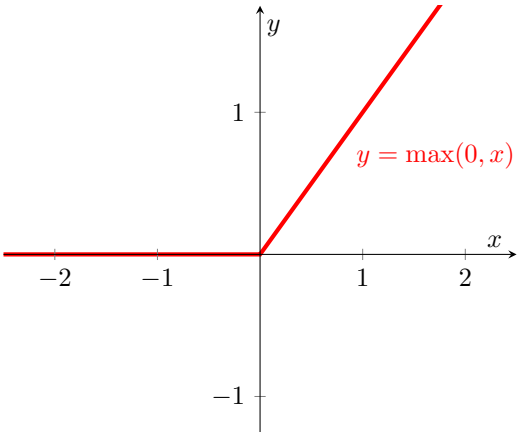
(a) Heaviside step function



(b) Sigmoid



(c) Hyperbolic tangent (tanh)



(d) Rectified linear unit (ReLU)

Figure 7: Some activation functions

The universal approximation theorem states that there exists a feedforward neural network, with at least one hidden layer and with any "squashing" activation function, approximating any Borel measurable function from one finite-dimensional space to another arbitrary well [24]. The sigmoid and hyperbolic tangent are typical examples of "squashing" activation functions. Note that set of Borel measurable functions is quite large, including for example every continuous function supported on a compact subset of $\mathbb{R}^n$. Later same approximation property is proved for a wider class of activation functions, including the rectified linear unit. However, even if the neural network can represent a given function very precisely, there is no guarantee that the training algorithm will learn the parameters of such a network. It is typical to use

a neural network with more than one hidden layer because, in many circumstances, parameters for such networks are easier to learn. Moreover, the deeper network might require fewer parameters in total because one hidden layer network might require a much larger hidden layer than layers in the deeper network.

3.1 Learning the parameters of a neural network

The goal in supervised deep learning is to find such parameters $\boldsymbol{\theta}$ that a neural network $\mathcal{NN}_{\boldsymbol{\theta}}$ approximate relation between input and output variables $\boldsymbol{x}$ and $\boldsymbol{y}$ following true data distribution p_{data}. The following quantity, known as the *risk*, measures how well this goal is achieved with respect to per-example loss function L:

$$J^*(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim p_{\text{data}}} L(\mathcal{NN}_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}).$$

Some choices for per-example loss function L are (weighted) ℓ^1 or ℓ^2 difference between $\mathcal{NN}_{\boldsymbol{\theta}}(\boldsymbol{x})$ and $\boldsymbol{y}$.

If the true distribution p_{data} is known, finding parameters $\boldsymbol{\theta}$, minimizing the risk, is optimization task. In the context of machine learning, only a training set of samples is available instead of entire data distribution. The simplest way to convert a machine learning problem back into an optimization problem is to minimize the expected loss on the training set. Replacing the true distribution p_{data} with the empirical distribution $\hat{p}_{\text{data}}$, defined by the training data, gives the *empirical risk* as the cost function $J(\boldsymbol{\theta})$ to minimize:

$$J(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim \hat{p}_{\text{data}}} L(\mathcal{NN}_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}) = \frac{1}{m}\sum_{i=1}^{m} L(\mathcal{NN}_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}), \boldsymbol{y}^{(i)}),$$

where m is the number of training samples. This is high dimensional optimization problem and has thus some challenges such as local minima.

There are some reasons why solving a machine learning problem is not typically just minimization of the empirical risk. Because neural networks are very flexible models, empirical risk minimization might lead to *overfitting*; the network starts to memorize training data instead of approximating the wanted relation. At some point, learning parameters that still reduce empirical risk will raise the corresponding risk. Regularization is needed to prevent overfitting. One way to regularize a neural network is to use a cost function with a regularization term $\Omega(\boldsymbol{\theta})$, encoding prior information. In some cases, another reason not to use the empirical risk as cost function is the per-example loss function unsuitable for optimization. This is solved by replacing unsuitable loss function with better behaving surrogate loss function. An example of a surrogate loss function is replacing the 0-1 loss with the negative log-likelihood in a classification task.

An optimization algorithm used to minimize a cost function $J(\boldsymbol{\theta})$ is typically based on the gradient of the cost function $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ computed using back-propagation algorithm, see section 3.1.1. As the cost is usually mean over the large training set, it is computationally expensive to compute gradient multiple times. Commonly, the gradient is estimated as the mean over a random sample of training data. These samples are called *minibatches*. Using larger batch size, amount of training examples in the minibatch, gives a better estimate for the gradient but small batches can offer a regularizing effect and help avoid getting stuck on local minima during optimization. Traditionally optimization algorithms that use only a single example at the same time were called *stochastic* and ones using larger samples minibatch or minibatch stochastic methods. Now it is common to call these all stochastic simply. The canonical example of a stochastic method is stochastic gradient descent, taking iteratively minibatch estimated gradient steps with predefined steplength, which is commonly called *learning rate* in the context of deep learning.

It can be slow to learn the parameters with stochastic gradient descent and hard to choose appropriate learning rate for the algorithm. More complex optimization algorithms are developed to make learning faster and easier. One common thing is to use the information of all gradients until the current iteration instead of just gradient computed in the current iteration. This gradient information is stored in weighted average, called *momentum* or *moment*. Typically new gradient is added to this average and all previous

ones, contained in average, are multiplied with some parameter smaller than 1. This causes momentum to be an exponential moving average, where the older the gradient is, the smaller the corresponding weight is. Another idea to ease the learning is to use adaptive learning rates for the individual model parameters during the training. Adam, a popular optimization algorithm using moments and adaptive learning rates, is presented detailed in algorithm 1. Adam computes exponential moving averages of gradients and squares of gradients, estimates of the 1st moment (the mean) and the 2nd raw moment (the uncentered variance) of the gradient [28]. As they are both initialized as zero vectors, Adam provides bias-correction to reduce the bias of moments towards zero.

Algorithm 1 Adam [28], adaptive moment estimation based stochastic optimization algorithm

Require: α, steplength
Require: $\beta_1, \beta_2 \in [0, 1)$ exponential decay rates for the moment estimates (default values 0.99 and 0.999 respectively)
Require: δ, small constant used for numerical stability (suggested default: 10^{-8})
Require: $J(\boldsymbol{\theta})$, stochastic objective function with parameter $\boldsymbol{\theta}$
Require: $\boldsymbol{\theta}_0$, initial parameter vector
Initialize variables the algorithm uses:
$\boldsymbol{m}_0 \leftarrow 0$
$\boldsymbol{v}_0 \leftarrow 0$
$t \leftarrow 0$
while stopping criterion not met **do**
  $t \leftarrow t + 1$
  $\boldsymbol{g}_t \leftarrow \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_{t-1})$
  Update biased first and second moment estimates:
  $\boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-1} + (1 - \beta_1)\boldsymbol{g}_t$
  $\boldsymbol{v}_t \leftarrow \beta_2 \boldsymbol{v}_{t-1} + (1 - \beta_2)\boldsymbol{g}_t \odot \boldsymbol{g}_t$
  Compute bias-corrected first and second moment estimates:
  $\hat{\boldsymbol{m}}_t \leftarrow \boldsymbol{m}_t / (1 - \beta_1^t)$
  $\hat{\boldsymbol{v}}_t \leftarrow \boldsymbol{v}_t / (1 - \beta_2^t)$
  Update parameters, all operations applied element-wise:
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \alpha \hat{\boldsymbol{m}}_t / (\sqrt{\hat{\boldsymbol{v}}_t} + \delta)$
end while

3.1.1 Back-propagation algorithm

The back-propagation algorithm [42] is used for efficient computation of the gradients of the cost function with respect to the parameters of a neural network θ. These gradients are needed for the optimization algorithm to update parameters θ such that the value of the cost function J decreases. The execution of the back-propagation algorithm requires calculating forward propagation first. During the forward propagation information flows from input to output one layer at a time, see algorithm 2, and then in back-propagation gradient information flows in the opposite direction by utilizing the chain rule of calculus. Each gradient of cost function indicates how the corresponding variable should change to reduce the cost, and this can be computed from the information on how the next variable should change to reduce the cost. The computation of the gradient is typically done over a minibatch of multiple training examples, but the back-propagation algorithm is presented here for the batch size of one for simplicity. This section focuses on the back-propagation algorithm for feedforward neural network, with the structure like in equation 33. Some considerations about back-propagation for CNNs provides in section 3.2.

Algorithm 2 Forward propagation for a feedforward neural network [19]

Require: Network depth, l
Require: $W^{(i)}, i \in \{1, ..., l\}$, the weight matrices of the model
Require: $\boldsymbol{b}^{(i)}, i \in \{1, ..., l\}$, the bias parameters of the model
Require: $\boldsymbol{x}$, the input of the network
Require: $\boldsymbol{y}$, the target output
1: $\boldsymbol{h}^{(0)} = \boldsymbol{x}$
2: **for** $k = 1, ..., l$ **do**
3: $\quad \boldsymbol{a}^{(k)} = W^{(k)}\boldsymbol{h}^{(k-1)} + \boldsymbol{b}^{(k)}$
4: $\quad \boldsymbol{h}^{(k)} = \sigma(\boldsymbol{a}^{(k)})$
5: **end for**
6: $\hat{\boldsymbol{y}} = \boldsymbol{h}^{(l)}$
7: $J = L(\hat{\boldsymbol{y}}, \boldsymbol{y}) + \lambda\Omega(\theta)$

The fact, that neural networks are compositions of multiple functions, enables utilizing of the chain rule of calculus in the back-propagation algorithm. For this purpose, it is useful to distinguish pre-activation of the layer k, $\boldsymbol{a}^{(k)} = W^{(k)}\boldsymbol{h}^{(k-1)} + \boldsymbol{b}^{(k)}$, and output $\boldsymbol{h}^{(k)} = \sigma_k(\boldsymbol{a}^{(k)})$ of the layer k. The chain rule of calculus states for the variables $\boldsymbol{h}^{(k-1)} \in \mathbb{R}^n, \boldsymbol{a}^{(k)} \in \mathbb{R}^m, \boldsymbol{a}^{(k)} = g(\boldsymbol{h}^{(k-1)})$ and value of the cost function $J \in \mathbb{R}$ the following relation

$$\frac{\partial J}{\partial h_i^{(k-1)}} = \sum_{j=1}^{m} \frac{\partial J}{\partial a_j^{(k)}} \frac{\partial a_j^{(k)}}{\partial h_i^{(k-1)}} \quad \text{for each } i \in \{1, ..., n\}. \tag{34}$$

The value of the cost function J depends on the output of the network $\hat{\boldsymbol{y}}$ and target output $\boldsymbol{y}$ in a way per-example loss function L defines and possibly also on regularizer $\Omega(\theta)$. Relation 34 written in vector notation is

$$\nabla_{\boldsymbol{h}^{(k-1)}} J = \left(\frac{\partial \boldsymbol{a}^{(k)}}{\partial \boldsymbol{h}^{(k-1)}}\right)^T \nabla_{\boldsymbol{a}^{(k)}} J, \tag{35}$$

which is $\nabla_{\boldsymbol{h}^{(k-1)}} J = W^{(k)T} \nabla_{\boldsymbol{a}^{(k)}} J$ if $\boldsymbol{a^{(k)}} = W^{(k)}\boldsymbol{h}^{(k-1)} + (\boldsymbol{b}^{(k)})$, i.e. g is linear (affine) function presented by matrix $W^{(k)}$ (and vector $\boldsymbol{b}^{(k)}$.) Equation 35 provide way to propagate the gradient of the cost function with respect to layer's pre-activations to the gradient of the cost function with respect to preceding layer's outputs. Since activation function σ_k is applied element-wise, propagation of the gradient of the cost function J on layer's output $\boldsymbol{h}^{(k)}$ back to the gradient on pre-activation $\boldsymbol{a}^{(k)}$ is just following application of one dimensional chain rule:

$$\frac{\partial J}{\partial a_i^{(k)}} = \frac{\partial J}{\partial h_i^{(k)}} \frac{\partial h_i^{(k)}}{\partial a_i^{(k)}} = \frac{\partial J}{\partial h_i^{(k)}} \sigma_k'(a_i^{(k)}) \quad \text{for each } i \in \{1, ..., n\}.$$

It can be written in vector notation as

$$\nabla_{\boldsymbol{a}^{(k)}} J = \nabla_{\boldsymbol{h}^{(k)}} J \odot \sigma'(\boldsymbol{a}^{(k)}). \tag{36}$$

Equations 35 and 36 shows how to propagate the gradients of the cost function back from the output of the network. These gradients and the information computed in the forward propagation can be used to compute gradient with respect to the weights $W^{(k)}$ (and the biases $\boldsymbol{b}^{(k)}$) using again the chain rule of calculus:

$$\frac{\partial J}{\partial W_{i,j}^{(k)}} = \frac{\partial J}{\partial a_i^{(k)}} \frac{\partial a_i^{(k)}}{\partial W_{i,j}^{(k)}} = \frac{\partial J}{\partial a_i^{(k)}} \frac{\partial [\sum_{k=1}^{n} W_{i,k}^{(k)} h_k^{(k-1)} + b_i^{(k)}]}{\partial W_{i,j}^{(k)}} = \frac{\partial J}{\partial a_i^{(k)}} h_j^{(k-1)}$$

for each $i \in \{1, ..., n\}$ and $j \in \{1, ..., m\}$. Same in vector notation is

$$\nabla_{W^{(k)}} J = \nabla_{\boldsymbol{a}^{(k)}} J \, \boldsymbol{h}^{(k-1)T}. \tag{37}$$

Gradient $\nabla_{\boldsymbol{b}^{(k)}} \boldsymbol{a}^{(k)}$ is just the identity. The back-propagation algorithm based on equations 35, 36 and 37 is summarized in algorithm 3.

Algorithm 3 Back-propagation for a feedforward neural network [19]

Require: Computation of the forward propagation, presented in algorithm 2

1: $\boldsymbol{g} \leftarrow \nabla_{\hat{\boldsymbol{y}}} J = \nabla_{\hat{\boldsymbol{y}}} L(\hat{\boldsymbol{y}}, \boldsymbol{y})$
2: **for** $k = l, l-1, ..., 1$ **do**
Convert the gradient on the layer's output into a gradient on the pre-activation (element-wise multiplication if σ is element-wise):
3: $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{a}^{(k)}} J = \boldsymbol{g} \odot \sigma'(\boldsymbol{a}^{(k)})$
Compute gradients on weights and biases (including the regularization term, where needed):
4: $\nabla_{W^{(k)}} J = \boldsymbol{g} \, \boldsymbol{h}^{(k-1)T} + \lambda \nabla_{W^{(k)}} \Omega(\theta)$
5: $\nabla_{\boldsymbol{b}^{(k)}} J = \boldsymbol{g} \; + \lambda \nabla_{\boldsymbol{b}^{(k)}} \Omega(\theta)$
Propagate the gradients w.r.t. the next lower-level hidden layer's outputs:
6: $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{h}^{(k-1)}} J = W^{(k)T} \boldsymbol{g}$,
7: **end for**

3.1.2 Batch normalization

Since neural networks are typically compositions of multiple functions, a small effect from changes to parameters in one layer can amplify significantly in subsequent layers. Still, the back-propagation algorithm computes gradients for each layer supposing other layers do not change and then update all the layers in practice simultaneously. Thus these updates can lead to unexpected results. The distributions of the layers' inputs might change drastically during the training. This can complicate the training by forcing the layers to adapt to the new distribution continuously. Batch normalization (BN) [25] is method developed to stabilize input distributions of the layers. It allows the use of higher learning rates and makes training less sensitive to the initialization of the parameters of the network. Batch normalization also helps the training of networks with sigmoid and hyperbolic tangent activation functions, because it keeps their inputs closer to the interval with derivatives not too small.

In batch normalization each dimension of d-dimensional input $\boldsymbol{x} = (x_1, ..., x_d)$ is normalized:

$$\widehat{x}_j = \frac{x_j - \mathbb{E}[x_j]}{\sqrt{\mathrm{Var}[x_j]}}.$$

To remain expressiveness of the network learnable parameters γ_j and β_j are introduced for scaling and shifting of the normalized value:

$$y_j = \gamma_j \widehat{x}_j + \beta_j.$$

This make it possible to undo normalization if it is preferred, When training neural network with stochastic optimization it is practical to use only samples of current mini-batch $\mathcal{B}$ (of size m) to get following estimates for $\mathbb{E}[x_j]$ and $\mathrm{Var}[x_j]$:

$$\mu_{\mathcal{B},j} = \frac{1}{m} \sum_{i=1}^{m} x_j^{(i)} \quad \text{and} \quad \sigma_{\mathcal{B},j} = \frac{1}{m} \sum_{i=1}^{m} (x_j^{(i)} - \mu_{\mathcal{B},j})^2.$$

Steps of batch normalization are differentiable and thus suitable for back-propagation algorithm. For detailed information, see article [25].

3.2 Convolutional neural networks

Convolutional neural networks (CNNs) are simply neural networks that use discrete convolution in place of general linear matrix operation in some of their layers. Convolution is an operation that uses knowledge of the known grid-like topology of its input. Images are an example of such input that consists of a 2-D grid of pixels, and it is known that neighboring pixels are usually related. As stated in introduction CNNs has become state-of-the-art in many computer vision tasks, including classification and segmentation [43, 18], and also achieved impressive performance solving imaging-related inverse problems [1, 22, 23, 26, 34].

From the mathematical perspective discrete 2D convolution for $I \in \mathbb{R}^{\mathbb{Z}\times\mathbb{Z}}$ and $K \in \mathbb{R}^{\mathbb{Z}\times\mathbb{Z}}$, more precisely I and K are mapping from $\mathbb{R}^{\mathbb{Z}\times\mathbb{Z}}$ to $\mathbb{R}^{\mathbb{Z}\times\mathbb{Z}}$, is operation defined as

$$(I * K)_{i,j} = \sum_{m,n\in\mathbb{R}^{\mathbb{Z}}} I_{m,n} K_{i-m+1,j-n+1} = \sum_{m,n\in\mathbb{R}^{\mathbb{Z}}} I_{i-m+1,j-n+1} K_{m,n} \quad \text{for } i,j \in \mathbb{R}^{\mathbb{Z}}. \tag{38}$$

In deep learning context I is called *input*, K *kernel* and the result of the convolution sometimes *feature map*. The fact that latter equality holds is known as the commutative property of convolution. For convolution to make sense, it is required that I and K are such that sum in equation 38 is finite. In practice, convolutions are computed for I and K, which has non-zero values only in a finite set, and this finite set is stored in matrix-like representation. This allows to sum over a finite set of indexes and makes to sum naturally finite. Following example visualizes computation of the first element of one type of 2D convolution for matrix input and kernel:

$$\left(\begin{bmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \end{bmatrix} * \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} \right)_{1,1} = 0.1 \cdot 22 + 0.2 \cdot 21 + 0.3 \cdot 12 + 0.4 \cdot 11 = 14.4.$$

Each element of the input and the kernel colored with the same color can be thought to lie in the same location and the result of the convolution obtained by summing up products of the overlapping elements. The order of the colors is different in the input and the kernel because the kernel is *flipped* in convolution. Flipping means the order of elements is reversed for all dimensions. Other elements of the output of the convolution are computed by translating the flipped kernel to another position overlapping the input, for visualization see figure 8. This example presents a type of convolution referred to as *valid* that restricts the output of the convolution to positions where the kernel lies entirely within the input. The position presented in this example is the first valid position and thus indexed as 1,1 according to typical indexing, where index increases from left to right and top to bottom.

The kernel is flipped in convolution operation to obtain commutative property. The commutative property is useful for writing proofs, but it is not usually an important property of neural network implementation. Furthermore, CNN does not commute even if the kernel is flipped. Many neural networks libraries implement cross-correlation, which is same as convolution but without flipping the kernel:

$$\sum_{m,n} I_{m,n} K_{i+m+1,j+n+1}, \tag{39}$$

but call it convolution. For learning algorithm, it does not matter if a neural network is defined with the flipped kernel or not, implementation with kernel flipping learns kernel that is flipped relative to kernel learned with implementation without flipping. In this text, both operations are called convolution and when it is relevant is kernel flipped, it is specified. Following example visualizes computation of convolution, without flipping the kernel:

$$\left(\begin{bmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \end{bmatrix} * \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} \right)_{1,1} = 0.1 \cdot 11 + 0.2 \cdot 12 + 0.3 \cdot 21 + 0.4 \cdot 22 = 18.6.$$

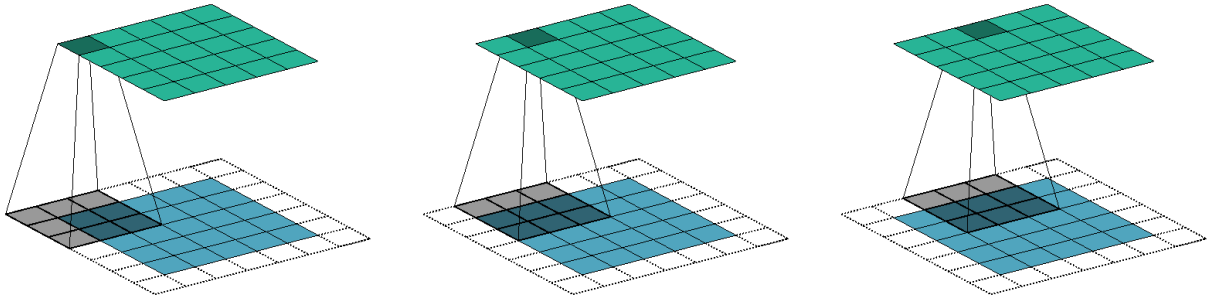


Figure 8: Visualization of a 2D (same) convolution. Bottom layers in subfigures represent zero padded inputs, white squares for padding and blue for input. Top layers present the output of the convolution. The shaded area in the bottom layers marks the position of the kernel and shaded area in the top layers element of output computed from this kernel position.

Let's consider different types of convolution operations. The valid convolution operation, presented earlier, has one drawback for deep neural networks. Size of the output is smaller than input, to be precise $o_d = i_d - k_d + 1$ for dimension d, where o_d denotes size of the output (in dimension d), i_d size of the input and k_d size of the kernel [16]. This could reduce the size of hidden layers close to the output layer of the convolutional network significantly, which might cause problems. To fix this issue, zero padding can be added to the input of the convolution operator to change the size of the output. One important case is convolution with an amount of zero padding keeping the dimension of the output the same as the dimension of the input. Example of the convolution with this padding known as *same* convolution is presented here (and in figure 8):

$$\left(\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \left[\begin{matrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \end{matrix}\right. & & & \\ 0 & & & & \\ 0 & & & & \end{matrix} * \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} \right)_{1,1} = 0.1 \cdot 11 + 0.2 \cdot 0 + 0.3 \cdot 0 + 0.4 \cdot 0 = 1.1.$$

Another example of a property that affects the size of the convolution is called *stride*, which performs downsampled convolution sampling only every s_i pixels in direction i of the output. Basic convolution has a stride of 1.

Convolutional layer and layer with general matrix multiplication are compared next. Important difference is that convolutional layer has typically *sparse interactions* (also known as *sparse connectivity*). This means that a single element of the convolutional layer's output is affected only by a small subset of layer's input and a single element of layer's input affects only a small subset of layer's output, for visualization see 8 and 9. A requirement for sparse interactions is that kernel is relatively small compared to the input of the convolution, which is a very common choice in the context of convolutional neural networks. Although units of consecutive layers have sparse interactions, unit form one layer can affect and be affected by each unit of some farther layer in deep CNN, see figure 10 for demonstration. While computing the output of the neural network, each connection between the units of the consecutive layers requires computational operation with a weight parameter attached to it. Therefore sparse connectivity reduces computational cost and memory requirements of a CNN compared to a respective fully connected neural network. Memory requirements of a CNN are reduced further by the property known as *parameter sharing*. It occurs as

reusing the same parameters of the kernel for multiple input positions, see figure 9.

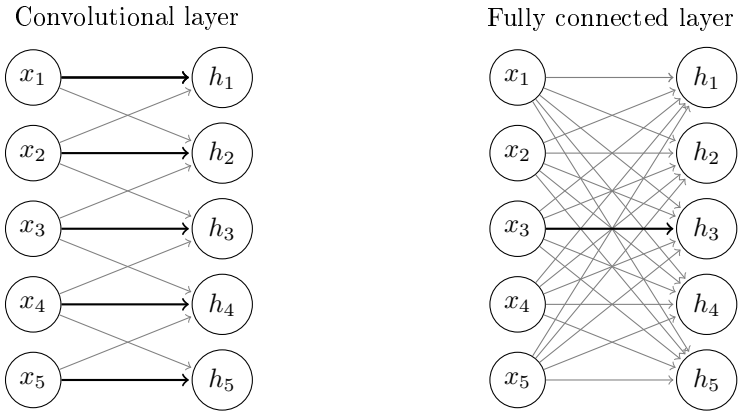


Figure 9: Visualization of differences between a convolutional layer and a fully connected layer, parameter sharing and sparse connectivity. Every connection in the convolutional layer (with kernel size 3×1) presented with black arrow uses the same weight parameter, but in the fully connected layer each connection has its own weight parameter. Sparse connectivity, in convolutional layers there are much fewer connections, arrows, between units of consecutive layers than in fully connected layers, which has a connection between every input and output unit. In convolutional layer unit x_3 affects only to units h_2, h_3 and h_4 and value of unit h_3 is only affected by values of units x_2, x_3 and x_3.

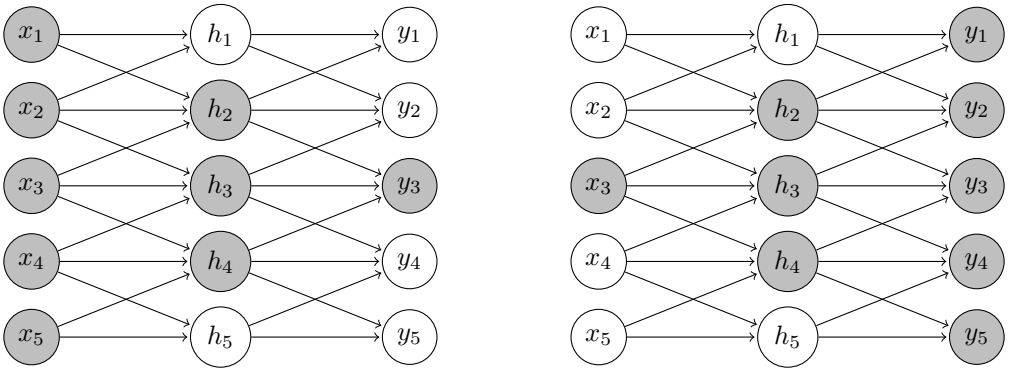
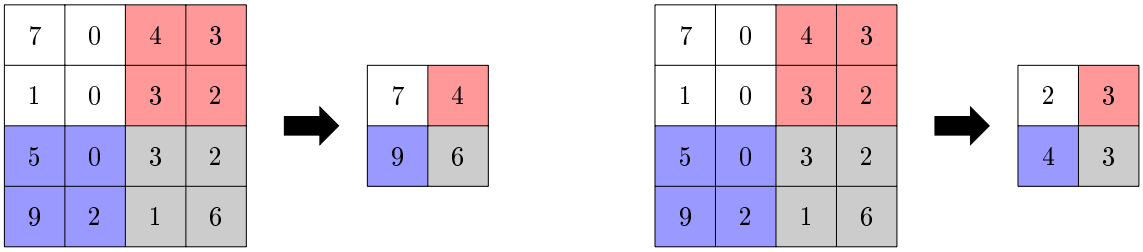


Figure 10: Even if units of consecutive layers are connected sparsely, single units may be indirectly connected to all units of another layer.

Convolutional neural networks usually employ operation called *pooling*. It is operation that computes summary statistics of input within rectangular neighbourhoods, defined by pooling window. For examples of pooling operations, see figure 11. It is also common that inputs and outputs of the convolutional layers have multiple channels of 2D inputs. Single output channel O_j for n input channels $\{I_i\}_{i \in n}$ are computed as

$$O_j = \sum_{i=1}^{n} I_i * K_i^{(j)},$$

where $K_i^{(j)}$ is the kernel with respect to *i:th* input channel and *j:th* output channel. RGB images are an example of input of CNN that have 3 channels. Typically also bias is added to output O_j.



(a) 2×2 max pooling with stride of 2

(b) 2×2 average pooling with stride of 2

Figure 11: Examples of pooling operations, 2×2 in the name of the operation refers to size of the pooling window [16].

The back-propagation of a convolutional neural network is considered next. Note first that there are no parameters to learn related to pooling operation. For max pooling forward propagation has to store witch element was maximum and information flows that path in the back-propagation. There are few changes to the iteration of the for-loop in the back-propagation algorithm, presented in the algorithm 3. Propagation of the gradients with respect to the next lower-level hidden layer's outputs, line 6 of the algorithm 3, involves matrix multiplication with the transpose of the weights. In practice transpose convolution is not either computed as matrix operation but as a type of convolution, see [16] for detailed information. Due to parameter sharing computation of the gradients on the weights, line 4 of the algorithm 3, differs for a convolutional layer from a fully connected layer. Following computation shows that for the convolutional layer the computation of the gradients on the weights is convolution operation, (here without flipping the kernel):

$$\begin{aligned}
\frac{\partial J}{\partial W^{(k)}_{m',n'}} &= \sum_{i,j} \frac{\partial J}{\partial a^{(k)}_{i,j}} \frac{\partial a^{(k)}_{i,j}}{\partial W^{(k)}_{m',n'}} = \sum_{i,j} \frac{\partial J}{\partial a^{(k)}_{i,j}} \frac{\partial (h^{(k-1)} * W^{(k)})_{i,j}}{\partial W^{(k)}_{m',n'}} \\
&= \sum_{i,j} \frac{\partial J}{\partial a^{(k)}_{i,j}} \frac{\partial}{\partial W^{(k)}_{m',n'}} \left[\sum_{m,n} W^{(k)}_{m,n} h^{(k-1)}_{i+m+1,j+n+1} + b^{(k)}_{i,j} \right] = \sum_{i,j} \frac{\partial J}{\partial a^{(k)}_{i,j}} h^{(k-1)}_{i+m'+1,j+n'+1} \\
&= \left(\nabla_{\boldsymbol{a}^{(k)}} J * h^{(k-1)} \right)_{m',n'}.
\end{aligned}$$

4 Methods

This section presents a method to solve the limited angle CT problems. This approach is highly motivated by the theory related to wavefront sets, presented in section 2 of this thesis. By the theorem 10, the wavefront set of the measurement object can be divided into visible and invisible parts. An appropriate model-based reconstruction method can recover the visible part of the wavefront set. Data-based methods were developed to estimate the invisible part of the wavefront set from the visible one and use this estimated wavefront set to obtain better reconstruction. The latter of these methods is called projection to (the space of the elements with) the fixed wavefront set and developing it was the main focus. A more detailed discussion about this projection is provided in subsection 4.1. To summarize, the methods developed in this thesis can be utilized to obtain a limited angle CT reconstruction by completing the following 4 steps:

Step 1: Obtain a model-based reconstruction $\tilde{\mathbf{f}}$ from the measurement.

Step 2: Extract the visible part of the wavefront set from the reconstruction $\tilde{\mathbf{f}}$.

Step 3: Estimate the invisible part of the wavefront set from the visible one.

Step 4: Project the reconstruction $\tilde{\mathbf{f}}$ such that the projection has the estimated wavefront set.

The algorithm above uses the developed methods for post-processing of the model-based reconstruction, but they could also be used in iterative algorithms.

4.1 Projection to the fixed wavefront set

Let us consider the space $X = L^2 \cap L^\infty(\Omega)$, where set $\Omega \subset \mathbb{R}^2$ is the measurement domain. Note that the model-based reconstruction can be less regular than the true object, and the domain X is chosen according to this fact. Suppose that a wavefront set S is given. One method to estimate it is provided in section 4.2.3. Let's denote $\mathcal{E}^2(\Omega, S) = \{\boldsymbol{y} \in \mathcal{E}^2(\Omega) : \mathrm{WF}(\boldsymbol{y}) = S\}$, i.e. $\mathcal{E}^2(\Omega, S) \subset \mathcal{E}^2(\Omega)$ is the subset of cartoon-like images with wavefront set S. A projection to wavefront set S, more precisely to the set $\mathcal{E}^2(\Omega, S)$, is defined to be the function $P_S : X \to \mathcal{E}^2(\Omega, S)$ such that for every $\boldsymbol{x} \in X$ it holds true

$$P_S^2(\boldsymbol{x}) = P_S(P_S(\boldsymbol{x})) = P_S(\boldsymbol{x}).$$

This property is known as *idempotency*, and it can also be formulated with the condition: the restriction $P_S|_{\mathcal{E}^2(\Omega,S)}$ is the identity mapping of the space $\mathcal{E}^2(\Omega, S)$. This operator is called projection since idempotency is an essential property of traditional projection that can be used for its characterization [29].

The idempotency condition does not yet specify, how the projection P_S maps elements from X to $\mathcal{E}^2(\Omega, S)$. Because it is a complex task to describe this relation explicitly with a closed form equation, a data-based approach is used. A parametric function $P_S(\cdot\,; \boldsymbol{\theta})$ is used as a proxy for the projection. Values for parameters $\boldsymbol{\theta}$ are set to approximate the relation between input and output variables $\boldsymbol{x}$ and $\boldsymbol{y}$ following the data distribution p_{data}. Therefore $P_S(\cdot\,; \boldsymbol{\theta}) : X \to \mathcal{E}^2(\Omega, S)$ is a function with parameters $\boldsymbol{\theta}$ minimizing the following constrained problem:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left\{ \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim p_{\text{data}}} \|P_S(\boldsymbol{x}; \boldsymbol{\theta}) - \boldsymbol{y}\|_2 \right\} \qquad \text{s.t. } P_S^2(\boldsymbol{x}; \boldsymbol{\theta}) = P_S(\boldsymbol{x}; \boldsymbol{\theta}) \text{ for every } \boldsymbol{x} \in X.$$

4.2 Practical implementation of the methods

A few words about the implementation in general are mentioned before describing the implementation of the steps more in details. After the model-based reconstruction is achieved in step 1, the other steps are

implemented in the shearlet domain. This is done because the decay properties of functions in the shearlet domain are related to the wavefront sets as stated in subsection 2.6. Implementation of steps 3 and 4 are based on the convolutional neural network architecture known as U-Net [41]. The detailed architectures are presented in figures 15 and 16. These U-Net based networks used in this thesis were implemented with the python framework called PyTorch [38]. Training of the networks was completed with a laptop having NVIDIA GeForce MX150 GPU, which has 2 GB of memory. This set limits to the size of the networks. Figure 12 presents the workflow of the method for a simple example target, the characteristic function of a disc multiplied with 0.5 to make it represent a sample from data generating distribution.

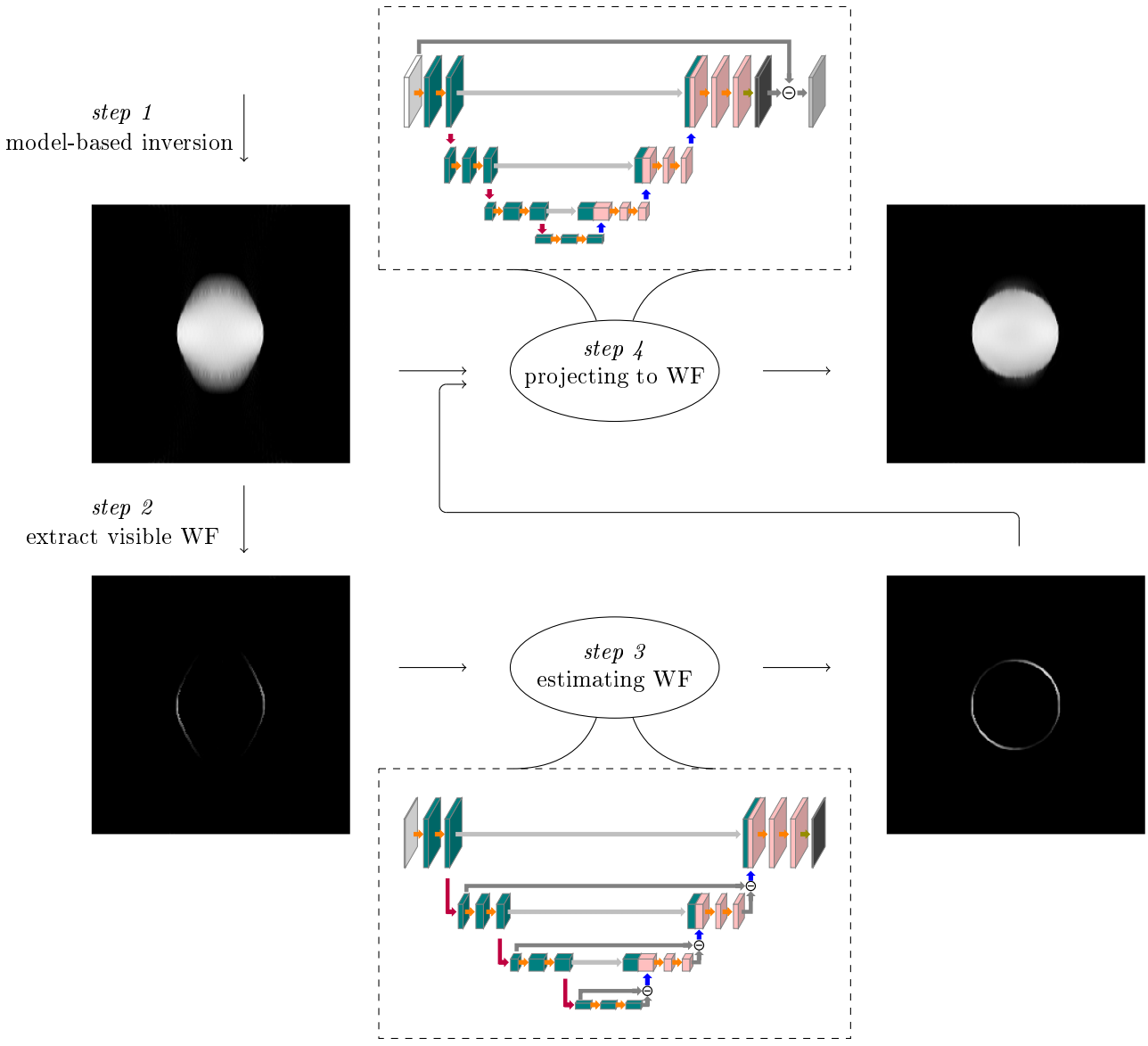


Figure 12: Schematic workflow of the method. Outputs of the steps are presented in image domain since it is demonstrative. More detailed figures of the network architectures are provided in figures 15 and 16.

4.2.1 Step 1: Model-based reconstruction

The projector and wavefront set estimator networks require limited angle CT reconstructions images for their inputs. In this thesis, these reconstructions were obtained using total variation regularization [37] with positivity constraint, denoted here by TV_+. This reconstruction method solves the minimization problem

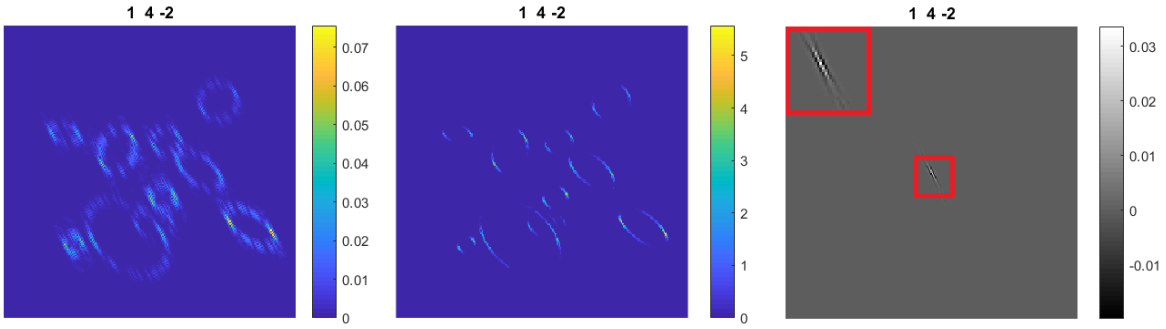
$$\mathbf{f}_{\mathrm{TV}_+} = \underset{\mathbf{f}\in\mathbb{R}^{N^2},\mathbf{f}\geq 0}{\operatorname{argmin}} \Big\{ \frac{1}{2}\|\mathcal{R}_\theta(\mathbf{f}) - \mathbf{m}\|_2^2 + \mu\|\mathbf{G}\mathbf{f}\|_1 \Big\},$$

where $\mathbf{m}$ are limited angle CT measurements, $\mathbf{G}$ is discrete gradient operator and μ the regularization parameter. The solution for positivity-constrained total variation regularization minimization problem was computed with primal-dual fixed-point algorithm [9]. Results section 5.4 shows that TV_+ gave better reconstructions than filtered back projection (FBP), a standard CT reconstruction method. In theory, FBP should recover the visible part of the wavefront set [30]. The visible wavefront set computed from the TV_+ reconstruction had still better quality for every example in simulation experiments. Thus TV_+ is more suitable input than FBP for both networks, the projector and the wavefront set estimator. A single TV_+ reconstruction is also quite fast to compute. This is important from the practical perspective when generating a large training set, where each example contains one reconstruction.

4.2.2 Step 2: Digital representation of wavefront sets

Since the method should project to fixed wavefront set, it is essential to have some digital representation for wavefront sets. Subsection 2.6 provides the connection between decay properties of the shearlet transform of a function and wavefront set of the corresponding function. Therefore shearlet based representation of wavefront sets is used in this thesis. Shearlet transforms were computed using the ShearLab [32] implementation using a compactly-supported shearlet system with 4 scales. The shearlet transform computed with this implementation has 49 subbands, corresponding to directional features at different scales. Each of these subbands has the size of the input image ($N \times N$) and thus the entire transform is an element of $\mathbb{R}^{N\times N\times 49}$. The number of scales was chosen according to two reasons. At first, the wavefront set is better defined by the shearlet coefficients of finer scales, which suggest choosing as many scales as possible. Since training the network requires lots of data, shearlet transforms, choosing implementation with more than 4 scales would have been impractical due to computational cost and memory requirements. Note for the sake of the theorem 21 that ShearLab implementation of shearlets has two vanishing moments. Remind that decaying of the shearlet transform at points not belonging to wavefront set is faster for shearlets with more vanishing moments.

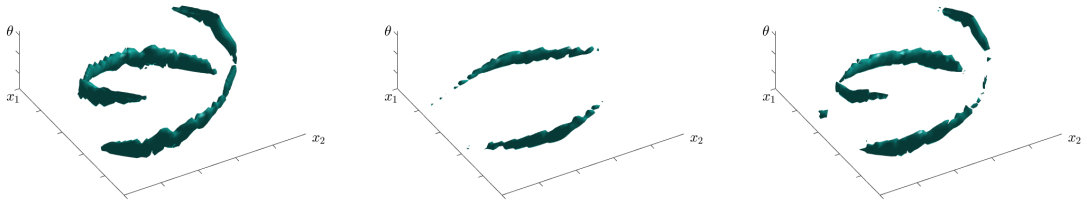
Figure 13 shows that the implementation of shearlet transform available in ShearLab has problems with defining wavefront set. For a direction determined by a shearing parameter, there are boundary points not belonging to the wavefront set with a higher value of shearlet transform than points belonging to the wavefront set. An algorithm called shearlet cleaner was used to overcome this problem. It is created by Samuli Peltonen for his Bachelor thesis for Samuli Siltanen. The algorithm is cleaning shearlets coefficients responsible for the wavefront set (computed with ShearLab) using a complex shearlet based edge detector presented in the article [27]. Even in this cleaned transform wavefront set for a range of directions are connected to one shearlet with one direction. This is true, because at scale 4 only 16 shearing parameters, subbands of the shearlet transform, are responsible for presenting wavefront set in every direction. Visualization of wavefront set representation based on cleaned shearlet coefficients is presented in figure 14. In fact, the networks presented in the next sections process the cleaned shearlet transforms instead of wavefront sets, which can be presented classifying each element of the shearlet transform. However, this finest scale of the cleaned shearlet transform is referred to as wavefront sets with abuse of terminology. The fact, that subbands of the cleaned shearlet transform at scale 4 are related



(a) Shearlab shearlet coefficients (b) cleaned shearlet coefficients (c) corresponding shearlet

Figure 13: Figure (a) shows part of the ShearLab implementation of the shearlet transform and figure (b) corresponding part of the cleaned shearlet transform. Indexes 1, 4 and -2 refer to vertical cone, scale parameter 4 and shear parameter -2. Boundary points clearly not belonging to the wavefront set in the direction of sheartlet presented in figure (c) has still significantly high value in figure (a)

to different directions of wavefront set, is important when distinguishing visible and invisible parts of the wavefront set.



(a) true (b) visible (and some more) (c) learned

Figure 14: Visualizations of the cleaned shearlet-based representation of wavefront set. Figure (a) presents the wavefront set of an ellipse, true object. Visible wavefront set determined by the angular range of the limited angle CT measurement is presented in figure (b). It contains little more wavefront set than just visible part because shearlet coefficients related to some shearing parameters contain information about visible and invisible parts. These directions are referred to as *semivisible*. Figure (c) shows an estimate of the wavefront set presented in figure (a) computed with a CNN.

4.2.3 Step 3 and 4: CNN architectures of the wavefront set estimator and the projector network

At first, deep learning methods for limited angle tomography motivating the chosen network architectures are discussed briefly. In the article [22], different CNN architectures performance on post-process FBP reconstructions was evaluated. U-Net type multiresolution architectures were found to perform better than single resolution architecture and learning in the wavelet domain instead of the image domain also improved the performance. The hybrid method learning in the shearlet domain, presented in paper [8], achieved even more impressive performance. This method used the directionality of shearlets to divide

the shearlet domain into the visible and invisible part and inferring only invisible part by deep learning. The used architecture, which was called PhantomNet, was also similar to U-Net but with added residual connections between consecutive layers at each scale. The input of the PhantomNet was obtained with an advanced model-based method that achieved better results than FBP.

Due to the success of other U-Net-based methods the architecture for both used networks, the wavefront set estimator $\mathrm{WF}_{\mathcal{NN}}$ and the projector network $P_{\mathcal{NN}}$, are based on U-Net [41]. It is a CNN that consists of contracting path and symmetric expansive path such that the overall architecture resembles a letter U, see figure 16. In contracting path feature maps are downsampled, by the factor of 2, to lower spatial dimension, but the number of channels is doubled after each downsampling. The expansive path does the opposite, feature maps are upsampled, by the factor of 2, to higher spatial dimension using transpose convolution, and after that the number of channels is halved. In each scale, both paths perform two operations consisting of convolution, batch normalization and ReLU activation function. Since U-Net is a so called fully convolutional network, it has no other types of layers, it can process inputs of varying spatial dimension. Another advantage of U-Net is that it can be trained successfully with a smaller amount of training data [41].

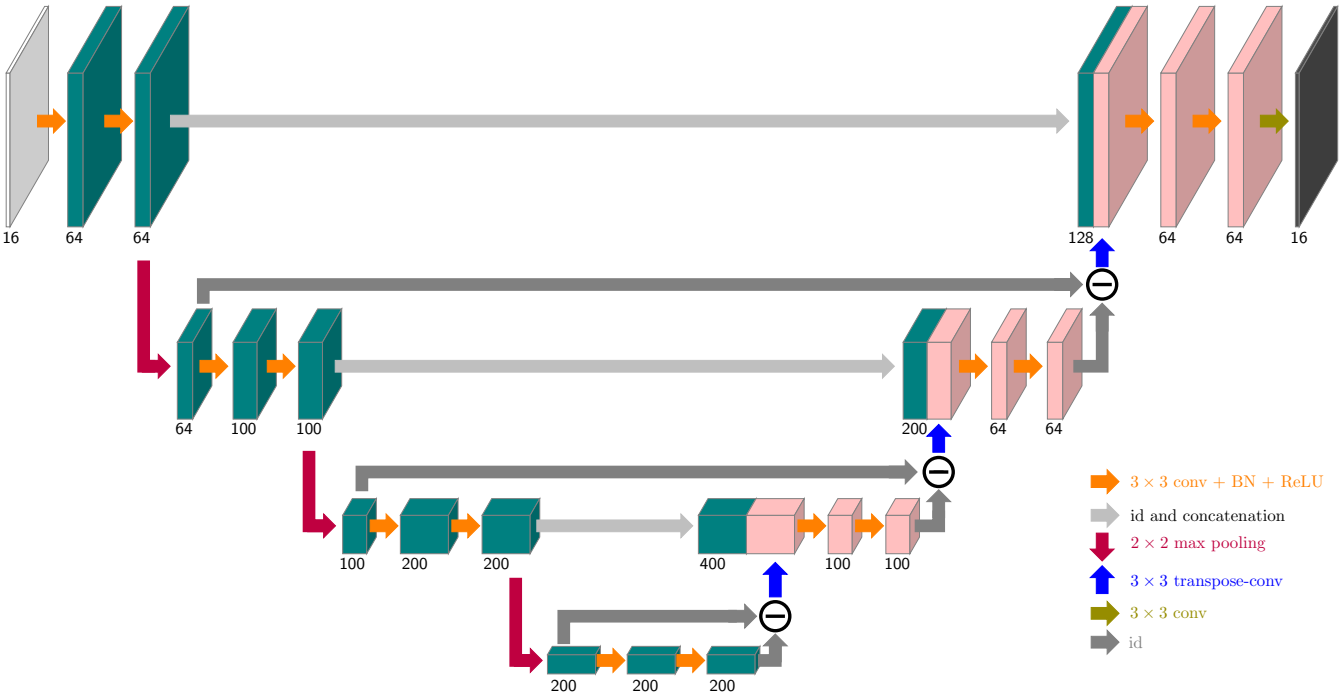


Figure 15: Diagram of the wavefront set estimator network ($\mathrm{WF}_{\mathcal{NN}}$) architecture. Boxes represent multi-channel feature maps, except the white one input and the black one output. The input of the network is cleaned shearlet representation of a visible wavefront set. It is presented in the entire wavefront set domain such that the subbands corresponding to invisible parts contains zeros.The number of channels is written below the boxes, and side length of the spatial dimension is reduced to half in each pooling operation. Operation $\ominus$ subtracts the output of a max pooling from the output of the layer preceding $\ominus$.

The fact that the wavefront set of the target can be divided into visible and invisible parts in limited angle tomography was the starting point for the development of methods for this thesis. Learning was

done in the shearlet domain because the decay properties of the shearlet transform of a function are related to its wavefront set. The finest scale of the digital shearlet transform represents these decay properties best. Therefore learning of the invisible information was completed in the finest scale instead of the entire shearlet domain, like with PhantomNet in article [8]. Another difference is that the wavefront set estimator network also infers visible directions of the wavefront set, but PhantomNet uses visible ones from the input. Results section 5.4 justifies this choice showing the output of the network estimates visible wavefront set better than the input. The quality of the input is worse in this thesis, since the angular range in the tomography task is significantly more limited. This also motivated adding semivisible directions to input, which deteriorate the quality of the input. The architecture of the wavefront set estimator network, $\mathrm{WF}_{\mathcal{NN}} : \mathbb{R}^{N\times N\times 16} \to \mathbb{R}^{N\times N\times 16}$, is presented in figure 15. It is obtained adding skip connections to make U-Net frame like in the article [23]. The final tuning of the architecture was done by comparing performance of different alternatives in little trials and one that seemed to be performing best was chosen.

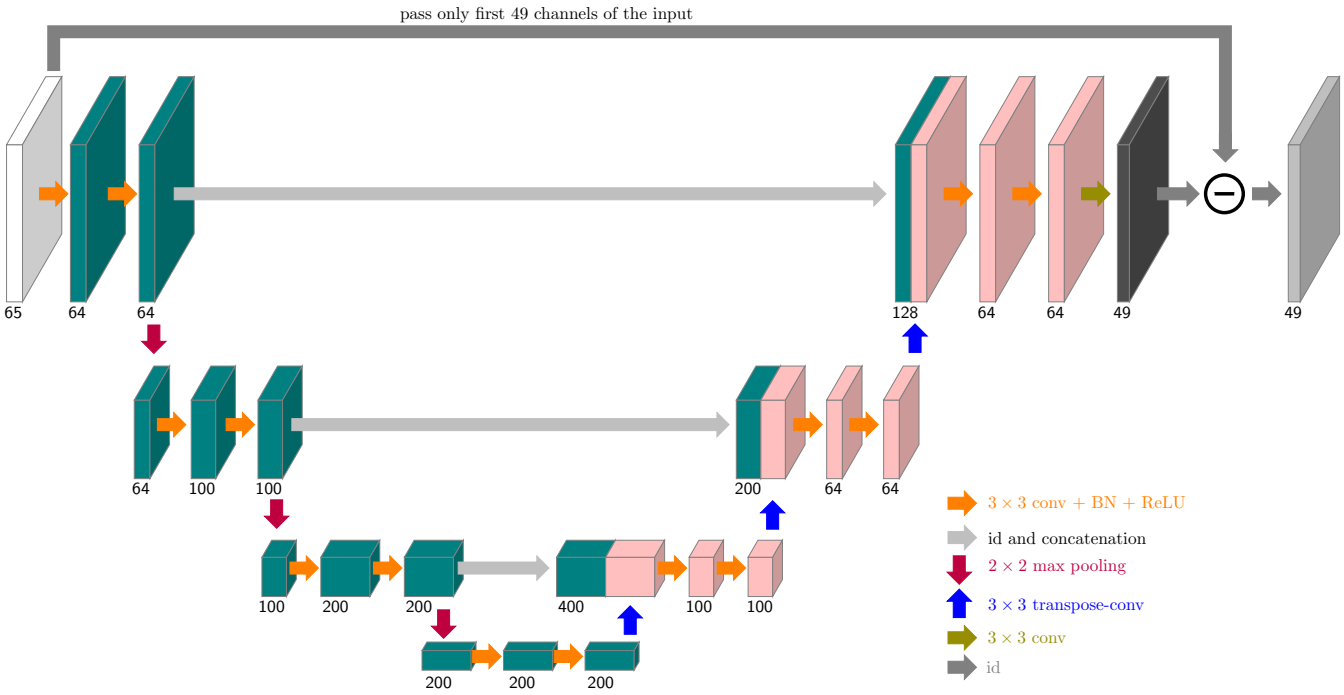
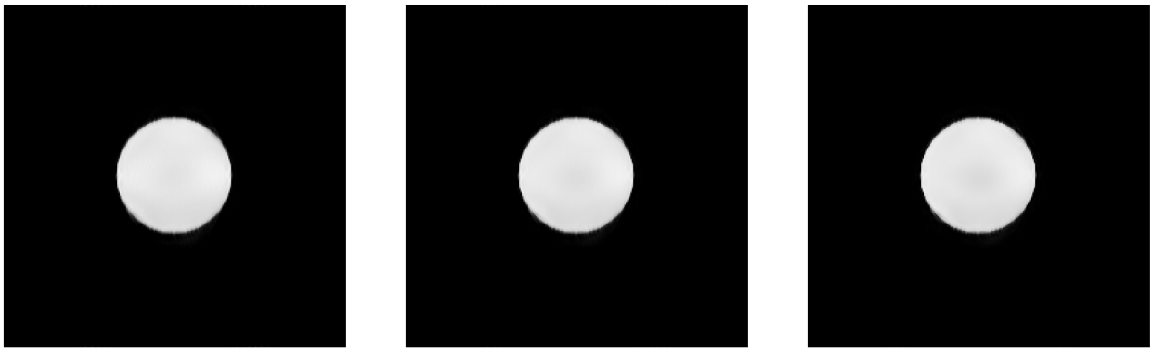


Figure 16: Diagram of the projector network ($P_{\mathcal{NN}}$) architecture, which is a residual U-Net. The subnetwork with the white box as input and the black box as output is an example of an U-Net. Boxes represent multi-channel feature maps, except the white one input and the gray one output. The input of the network is concatenation of the shearlet transform of a reconstruction and cleaned shearlet representation of a wavefront set to project to. The output is shearlet transform of the projected reconstruction. The number of channels is written below the boxes, and the side length of the spatial dimension is reduced to half in each pooling operation. Operation $\ominus$ subtracts the black box from the first 49 channels of the white one.

The projector network $P_{\mathcal{NN}} : \mathbb{R}^{N\times N\times 65} \to \mathbb{R}^{N\times N\times 49}$, presented in figure 16, is closely related to the methods to solve limited angle CT problem using modified U-Net, proposed in the articles [22] and [8]. Architecture for the projector network is very similar to one in the article [22], where a residual version of the U-Net was found the best performing CNN architecture type. Article [26] also proposes residual learning for

this kind of task where output is supposed to be close to input. There are two essential differences between the projector network and the network proposed in the article [22]. The projector network modifies shearlet coefficients instead of wavelet coefficients and its input also contains a representation of a wavefront set. Using shearlet transform as input is motivated by the success of the shearlet domain method in the article [8] and because the representation of the wavefront set is based on shearlets. Wavefront set is contained in the input because the idea of this method is to obtain better reconstructions by using information from the previous step. Since the input of the projector networks consists of a (shearlet transform of) reconstruction and wavefront set parts $\boldsymbol{x}$ and S, a notation $P_{\mathcal{NN}}(\boldsymbol{x}, S; \boldsymbol{\theta})$ is used to denote the corresponding output of the network parameterized by $\boldsymbol{\theta}$. Figure 17 shows an example of how well the projector network succeeds to fulfill the idempotency property, and it is further examined in the results section 5.4.



(a) `proj` (b) $\texttt{proj}^2$ (c) $\texttt{proj}^3$

Figure 17: Visualization of the idempotency property of the projector network. These projections are obtained by giving the wavefront set of true object as input to the projector network. Differences of ouputs between projecting once and more are quite small.

5 Experiments and results

5.1 Experimental scenarios

There were experiments with three types of 256×256 pixel targets:

1. The *ellipses dataset* consists of simulated targets of the form $\sum_{i=1}^{n} a_i \mathbb{1}_{E_i} / \max\{\sum_{i=1}^{n} a_i \mathbb{1}_{E_i}\}$, where each E_i is an ellipse, a_i uniformly sampled from interval $[0.2, 1]$. In 80% of the cases $n = 10$ and in the rest n is smaller. The division by $\max\{\sum_{i=1}^{n} a_i \mathbb{1}_{E_i}\}$ is done to rescale values of the target to the interval $[0, 1]$. This scaling is one of the choices that should make this dataset simpler and easier for deep learning. A simple dataset was preferred, because it should show with less effort if this method is suitable for severely ill-posed limited angle CT. Other choices to simplify dataset were done in the sampling of the ellipses E_i. Center points and sizes of the ellipses were chosen such that most of them should be entirely in the measurement domain. When sampling a new ellipse, it was resampled a few times if the ellipse overlapped with ellipses already sampled to this target. For the visualization of an example target see figure 18. Tomographic data was simulated (with `radon` function of Matlab) for three parallel beam measurement setups:

   (i) Ellipses-25° (low noise) has simulated $\mathcal{R}_{25^\circ}$ measurements with noise level 0.001, which means standard deviation of the normally distributed noise is 0.001 times the maximum value of the simulated measurement. This scenario is also referred to as Ellipses-25°, i.e. without the low noise specification. The angular step between consecutive measurement angles was 2° as it was in every type of Ellipse-measurement. The chosen opening angle ±25° is motivated by a breast tomosynthesis device [47]. Architectural choices for the used networks are mostly based on performance on the Ellipses-25° validation set.

   (ii) Ellipses-40° has simulated $\mathcal{R}_{40^\circ}$ measurements with the same noise level 0.001. The purpose of this data is to see how good reconstructions are obtained with little easier data.

   (iii) Ellipses-25° (high noise) has simulated $\mathcal{R}_{25^\circ}$ measurements with noise level 0.01. This data is generated to see how the method manages to handle noisy inputs.

2. The *smooth dataset* is named after the shape of the support of the simulated targets. The boundary of the support is a smooth curve generated by connecting 8 random control points in sequential order with cubic Bézier curves [44]. The corresponding image is smoothed with a Gaussian filter (convolution with Gaussian kernel) and then thresholded to obtain a smooth curve. The smoothing is done because the original curve is possibly discontinuous at the control points. The smooth targets were also sums of constants b_i times characteristic functions like ellipsoid dataset but with few differences. Instead of scaling values of the smooth targets were forced to interval $[0, 1]$ with the thresholding $\min\{1, \max\{0, \cdot\}\}$. Another difference is that other sets in the sum were subsets of the support generated first. The idea is that the shape of the support defines the overall structure of the target and these subsets add some finer details. The shapes of these subsets were quadrilateral, ellipse and smooth that was generated like the support but with only one control point. There were from 0 to 5 of each type of these subsets. Constant multiplier b_0 for the support was uniformly sampled from interval $[0.2, 0.5]$. Constants for the smooth subsets were uniformly sampled from the interval $[-0.5, -0.1]$ and for the quadrilateral and ellipse from $[-0.5, -0.1] \cap [0.1, 0.5]$. For the visualization of an example target see figure 18.

   Tomographic data for the smooth dataset (Smooth-40°) was simulated using the same fanbeam measurement set up as in *lotus root* measurement, see the paper [7]. Thus the network trained with this data is suitable for testing the performance with the real measurement. Measurement matrix A form file `Data256.mat`, referred in the paper [7], specifies full angle measurement with 3° degree

angular steps between measurement angles. A submatrix of this matrix A corresponding to $\pm 40°$ opening angle is used for measurement simulation and normally distributed noise with noise level 0.001 is added.

3. Lotus root data described in the paper [7] is used to check the generalization capabilities of the method to real measured data. Note that this data has only one sample and it is not used for training, but only for evaluation.

Simulated training sets had 1000 samples except Ellipses-25°, which had 3000 samples for wavefront estimation and 1000 samples for projection. Each simulated validation set had 50 samples and simulated test set 100 samples. In simulation experiments, targets and measurements were rotated in opposite directions. The purpose of this was to make the measurement model of the reconstruction algorithm less perfect to avoid *inverse crime* [37].

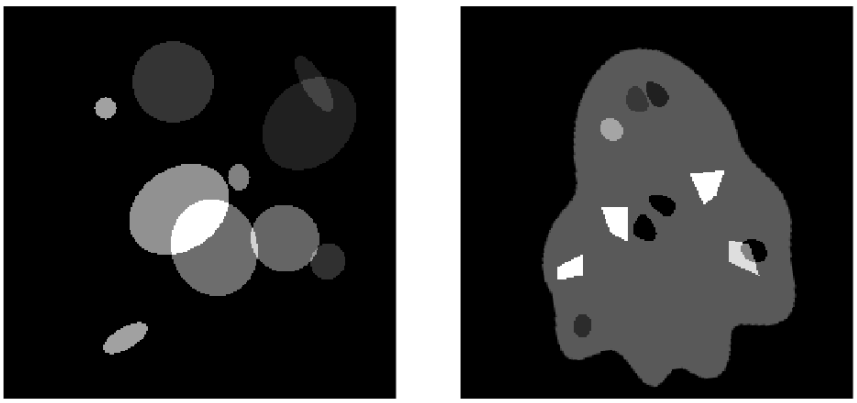


Figure 18: Example targets from the ellipses (left) and the smooth (right) datasets.

5.2 Training

Training of the networks was performed using PyTorch [38] with an Adam optimizer, see algorithm 1, and a learning rate of 10^{-3}. The chosen cost functions based on ℓ_1 per-example loss, since it suits for sparse shearlet domain and wavefront set information. Actually, weighted ℓ_1-loss was used for training the wavefront set estimator network $\mathrm{WF}_{\mathcal{NN}}$. Enough non-zero, larger than 0.1, elements of the target were weighted with $\lambda_1 > 1$. This weighting helped the network to find points belonging to the wavefront set, and possibly some false positives close to them also. However, it was preferred to not finding the wavefront set for a significant angle at all. The projector network $P_{\mathcal{NN}}$ was pushed towards idempotency by adding the regularization term $\|P^2_{\mathcal{NN}}(\boldsymbol{x}, S; \boldsymbol{\theta}) - P_{\mathcal{NN}}(\boldsymbol{x}, S; \boldsymbol{\theta})\|_{\ell_1}$ to the cost function. The corresponding regularization parameter λ_2 needed to be chosen quite small to avoid learning just identity mapping, especially in cases where the difference between input and output is not that large. Parameter values for λ_1 and λ_2 in different experimental scenarios are listed below:

- Ellipse-25° (low noise) : $\lambda_1 = 3$, $\lambda_2 = 5 \cdot 10^{-8}$,
- Ellipse-40°: $\lambda_1 = 2$, $\lambda_2 = 1 \cdot 10^{-8}$,
- Ellipse-25° (high noise): $\lambda_1 = 2$, $\lambda_2 = 5 \cdot 10^{-8}$,
- Smooth-40°: $\lambda_1 = 5$, $\lambda_2 = 1 \cdot 10^{-8}$.

The generalization capabilities of a neural network typically improve when the size of the training set increases. Because training sets for ellipse and smooth data has only 1000 samples, additional data augmentation technique was used to help the networks to converge to good local minimizers of the empirical risk. During the training spatial dimension of the input was reduced by sampling 200×200-patch and cropping each subband of the input according to it. Such on-the-fly sampling was found to provide effective regularization for similar training in the paper [8]. Note that for the validation (and testing) input with full spatial dimension was given to the network. The networks can handle data with varying spatial dimensions since they are fully convolutional.

5.3 Similarity measures

Three similarity measures, relative error (RE), the structural similarity index (SSIM) [45] and the Haar wavelet-based perceptual similarity index (HaarPSI) [40], were used for evaluating the quality of reconstructions. For a reconstruction $\mathbf{f}_{\text{rec}}$ and the target $\mathbf{f}$ relative error is computed as

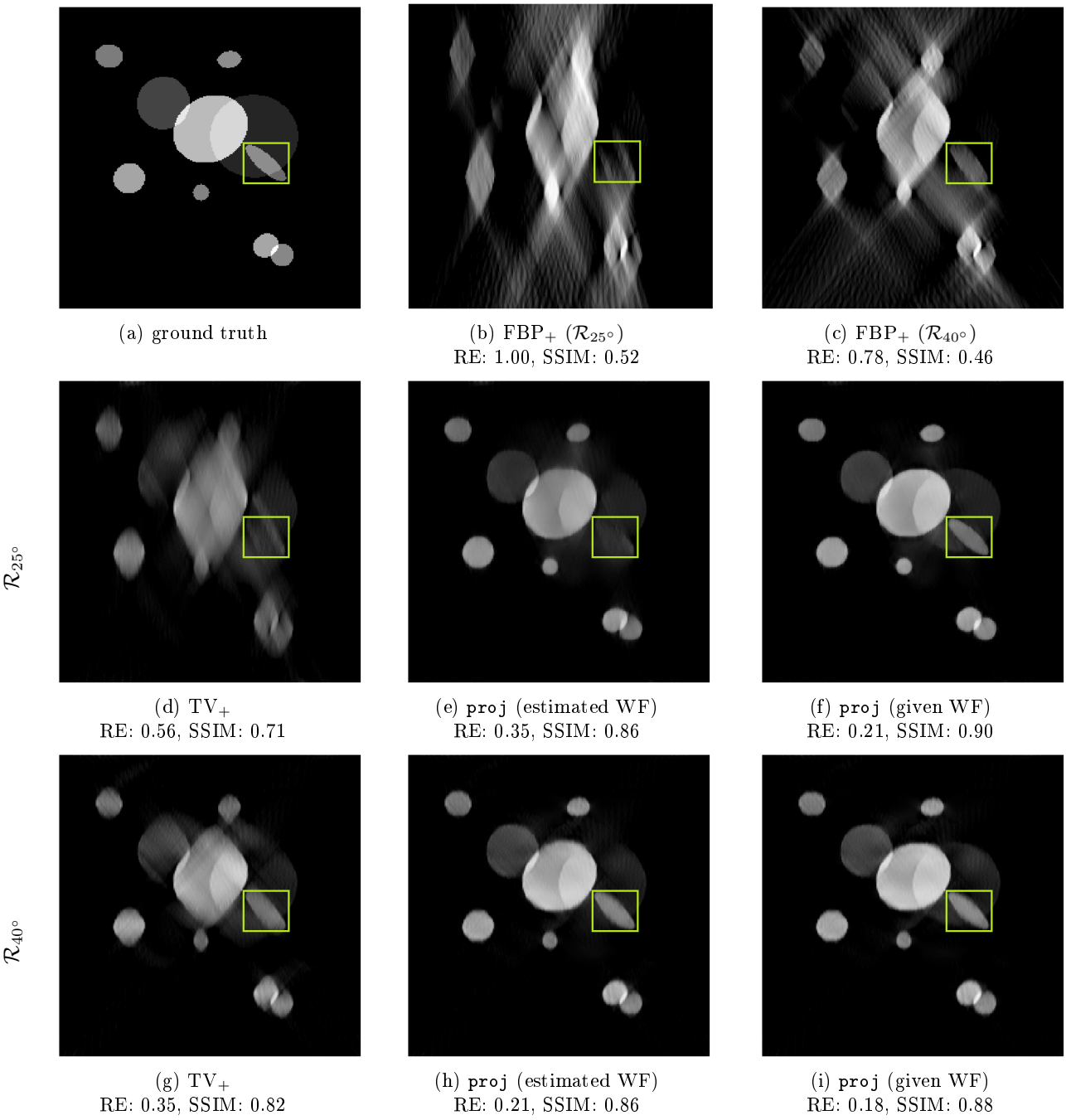
$$\|\mathbf{f}_{\text{rec}} - \mathbf{f}\|_2 / \|\mathbf{f}\|_2.$$

It measures actually difference instead of similarity, but the connection between these is clear: the smaller the difference, the more similar are the compared objects. SSIM and HaarPSI obtain values from intervals $[-1, 1]$ and $[0, 1]$ with the optimal value 1 but the optimal value for relative error is 0. SSIM and HaarPSI are designed for measuring image similarity, but relative error is suitable in more general cases. Therefore relative error with ℓ^1-norm (instead of ℓ^2-norm) was used for measuring the similarity of objects in the shearlet domain, which consists of many image-like subbands. Version with ℓ^1-norm (ℓ^1-RE) was chosen because many subbands of the shearlet domain contains sparse information, especially ones used for the wavefront set representation.

The similarity indices SSIM and HaarPSI are introduced next briefly. For more information see the corresponding papers [45] and [40]. SSIM is based on the computation of three components, called *luminance*, *contrast* and *structural* similarities. Luminance similarity compare (local) mean values, contrast similarity (local) variances, and structural similarity (local) covariance between reconstruction and the target image. Local SSIM values are obtained as a multiplicative combination of these three components where statistics are computed within a circular symmetric Gaussian weighting window. The overall index is the mean of the local indexes. Computation os SSIM was done using Matlab implementation with default choices. The HaarPSI utilizes the magnitudes of high-frequency Haar wavelet coefficients to define local similarities and low-frequency Haar wavelet coefficients to weight these similarities at specific locations in the image domain. The six discrete two-dimensional Haar wavelet filters used in the HaarPSI respond to horizontal and vertical edges on different frequency scales. Implementation of HaarPSI is available at `http://www.haarpsi.org`.

5.4 Results

At first, the terminology used in the tables and figures presenting results is explained. Scenarios Ellipse-25° and Ellipse-40° are also referred to as $\mathcal{R}_{25^\circ}$ and $\mathcal{R}_{40^\circ}$ respectively. All the tables present statistics in the form mean $\pm$ standard deviation. The last step of all the compared methods is forcing non-negativity by element-wise $\max\{0, \cdot\}$ operation. The usage of the non-negativity constraint motivates the subscripts in the naming FBP_+ and TV_+. The method using both developed networks to postprocess the TV_+ reconstruction is referred to as `proj`. The performance of the projector network with the true wavefront set given is also tested. This method is called `proj oracle` and is separated with a dashed line in the tables because it uses wavefront set information not obtained from the measurement data. On some occasions, these are both referred to as to `proj` but it is specified is the wavefront set estimated or given. Note that



(a) ground truth

(b) FBP_+ ($\mathcal{R}_{25°}$)
RE: 1.00, SSIM: 0.52

(c) FBP_+ ($\mathcal{R}_{40°}$)
RE: 0.78, SSIM: 0.46

(d) TV_+
RE: 0.56, SSIM: 0.71

(e) `proj` (estimated WF)
RE: 0.35, SSIM: 0.86

(f) `proj` (given WF)
RE: 0.21, SSIM: 0.90

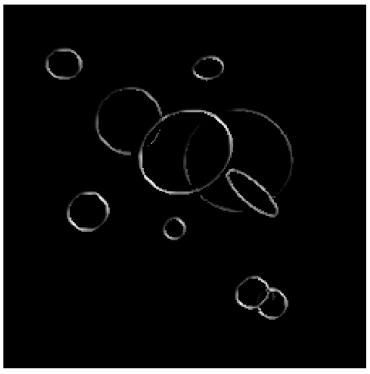
(g) TV_+
RE: 0.35, SSIM: 0.82

(h) `proj` (estimated WF)
RE: 0.21, SSIM: 0.86

(i) `proj` (given WF)
RE: 0.18, SSIM: 0.88

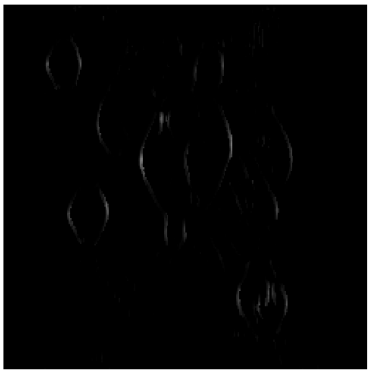
Figure 19: Visualization of the results for one test sample in scenarios Ellipse-25° and Ellipse-40°. The green rectangles in images are there to point ellipses hardest to reconstruct. For averaged similarity measures over the test set, see table 1.

in figures presenting reconstructions all the subfigures except the ones presenting FBP_+ are shown in the

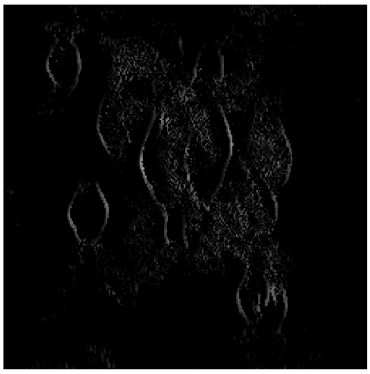


(a) ground truth

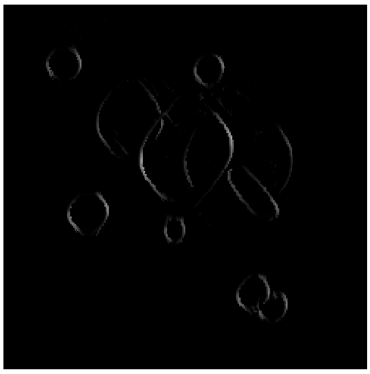
inputs



(b) $\mathcal{R}_{25^\circ}$ (low noise)
ℓ^1-RE: 1.12

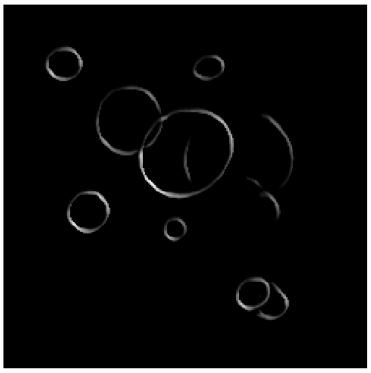


(c) $\mathcal{R}_{25^\circ}$ (high noise)
ℓ^1-RE: 3.16

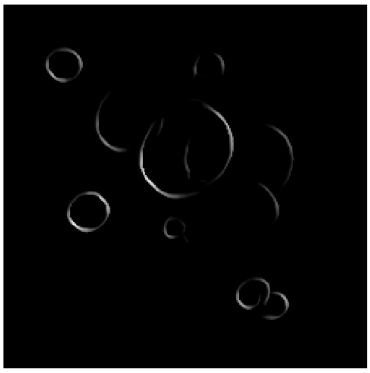


(d) $\mathcal{R}_{40^\circ}$
ℓ^1-RE: 0.89

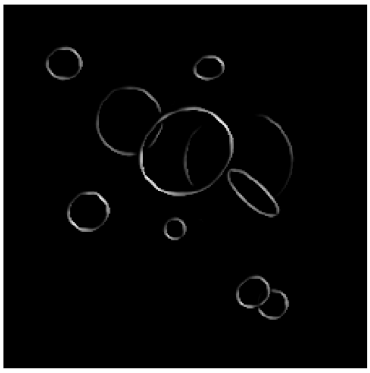
outputs



(e) $\mathcal{R}_{25^\circ}$ (low noise)
ℓ^1-RE: 0.77



(f) $\mathcal{R}_{25^\circ}$ (high noise)
ℓ^1-RE: 0.91



(g) $\mathcal{R}_{40^\circ}$
ℓ^1-RE: 0.47

Figure 20: Visualizations of the wavefront set estimation for the different Ellipse scenarios. For the inputs, subfigures b, c and d, errors are computed only for the visible parts of the wavefront sets. Statistics for the wavefront set estimation are provided in table 2.

same plotting window. Adjusting all subfigures to the plotting window of FBP_+ reconstructions would

have worsened the contrast for the more interesting reconstructions.

	$\mathcal{R}_{25^\circ}$			$\mathcal{R}_{40^\circ}$		
Method	**RE**	**SSIM**	**HaarPSI**	**RE**	**SSIM**	**HaarPSI**
FBP_+	1.01 ± 0.09	0.51 ± 0.06	0.13 ± 0.02	0.72 ± 0.04	0.50 ± 0.06	0.19 ± 0.03
TV_+	0.42 ± 0.07	0.75 ± 0.07	0.23 ± 0.03	0.28 ± 0.05	0.84 ± 0.05	0.31 ± 0.05
`proj`	$\mathbf{0.27 \pm 0.05}$	$\mathbf{0.88 \pm 0.04}$	$\mathbf{0.43 \pm 0.05}$	$\mathbf{0.19 \pm 0.03}$	$\mathbf{0.89 \pm 0.04}$	$\mathbf{0.51 \pm 0.05}$
`proj oracle`	0.17 ± 0.03	0.92 ± 0.03	0.61 ± 0.05	0.16 ± 0.02	0.91 ± 0.04	0.58 ± 0.04

Table 1: Comparison of reconstructions methods performance for Ellipse-25° and Ellipse-40° test sets of 100 samples. Example reconstructions are shown in figure 19 and for Ellipse-25° also in figures 21 and 22.

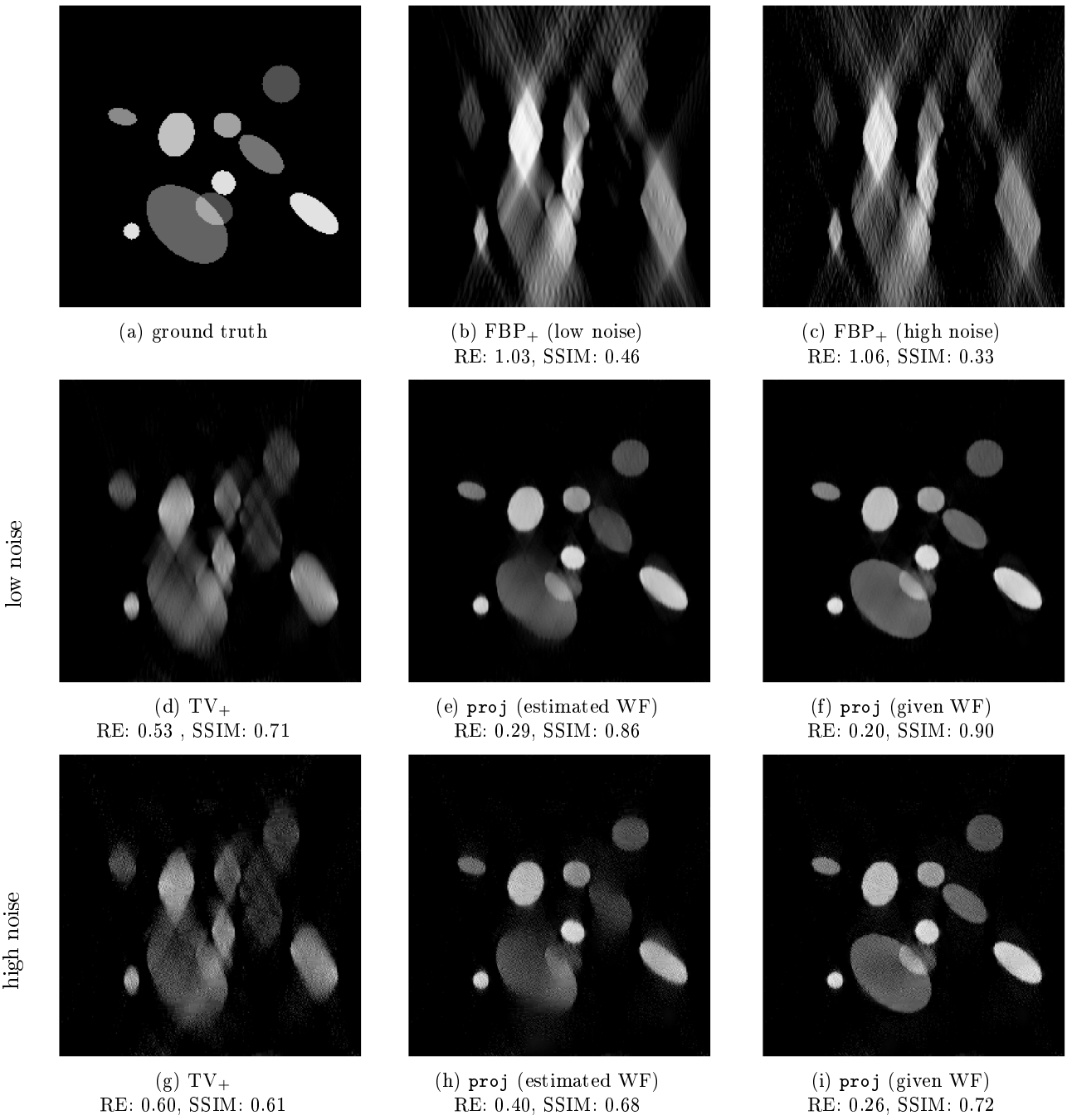
Test set performance statistics for Ellipse-25° and Ellipse-40° are presented in the table 1 and figure 19 shows example reconstructions for corresponding scenarios. Wavefront set estimation results for the same example target are provided in figure 20. Figure 19 shows how model-based reconstructions of ellipses are stretched in the vertical direction, where the wavefront set is unknown. The proposed method estimates this invisible wavefront set reasonably well for most of the ellipses. However, in scenario Ellipse-25° all the compared methods (except one with given WF) has problems reconstructing the ellipse, which is highlighted with a green rectangle in figure 19. This ellipse is quite horizontal, which makes only a small part of its wavefront set is visible for $\mathcal{R}_{25^\circ}$ measurement. Figures 19 and 20 show also problems with the reconstruction of the ellipse that is partially in the green rectangle. This represents the situation with problems of reconstructing weakly attenuating ellipse that intersects with much stronger attenuating ellipses. There is one more property of the projector network that can be seen in figures 19 and 21. Weak versions of the stretched part of ellipses in TV_+ are left to reconstruction after the projection with the network. Despite the described bad properties of the proposed method both tables and images show it still outperforms the model-based methods clearly. Reconstructions with the true wavefront set given show that the projector network can perform even better if the wavefront set estimation improves.

Experiment	ℓ^1-$\mathbf{RE}_{\text{input: visible WF}}$	ℓ^1-$\mathbf{RE}_{\text{output: visible WF}}$	ℓ^1-$\mathbf{RE}_{\text{output: entire WF}}$
Ellipse-25° (low noise)	1.01 ± 0.08	0.65 ± 0.10	0.80 ± 0.11
Ellipse-25° (high noise)	3.40 ± 1.24	0.76 ± 0.09	0.90 ± 0.08
Ellipse-40°	0.76 ± 0.08	0.45 ± 0.07	0.57 ± 0.08
Smooth-40°	1.04 ± 0.08	0.64 ± 0.6	0.85 ± 0.08

Table 2: Statistics for wavefront set estimation showing the estimate with neural network for visible wavefront set is better than the one extracted from the TV_+ reconstruction.

Performance on Ellipse-25° with different noise levels is presented in table 4 and single samples on figures 21 and 22. Table 3 shows the performance of the projector network trained with low noise data is comparable with the projector network trained with noisy data. Therefore results are presented for the network trained with low noise data, since it is practical to train as few neural networks as possible to obtain results. A different network is still used for the wavefront set estimation, which table 2 and figure 20 shows to be significantly harder in high noise case. Even if noisiness reduces the quality of all reconstructions the proposed method succeeds to improve the quality of model-based reconstruction notably.

Table 5 provides statistics for assessing how well the projector network, trained on Ellipse-25°, fulfills idempotency property in different noise levels. The used notation $\text{RE}_{i,j}$, $(i < j)$ stands for the relative



(a) ground truth

(b) FBP_+ (low noise)
RE: 1.03, SSIM: 0.46

(c) FBP_+ (high noise)
RE: 1.06, SSIM: 0.33

(d) TV_+
RE: 0.53 , SSIM: 0.71

(e) `proj` (estimated WF)
RE: 0.29, SSIM: 0.86

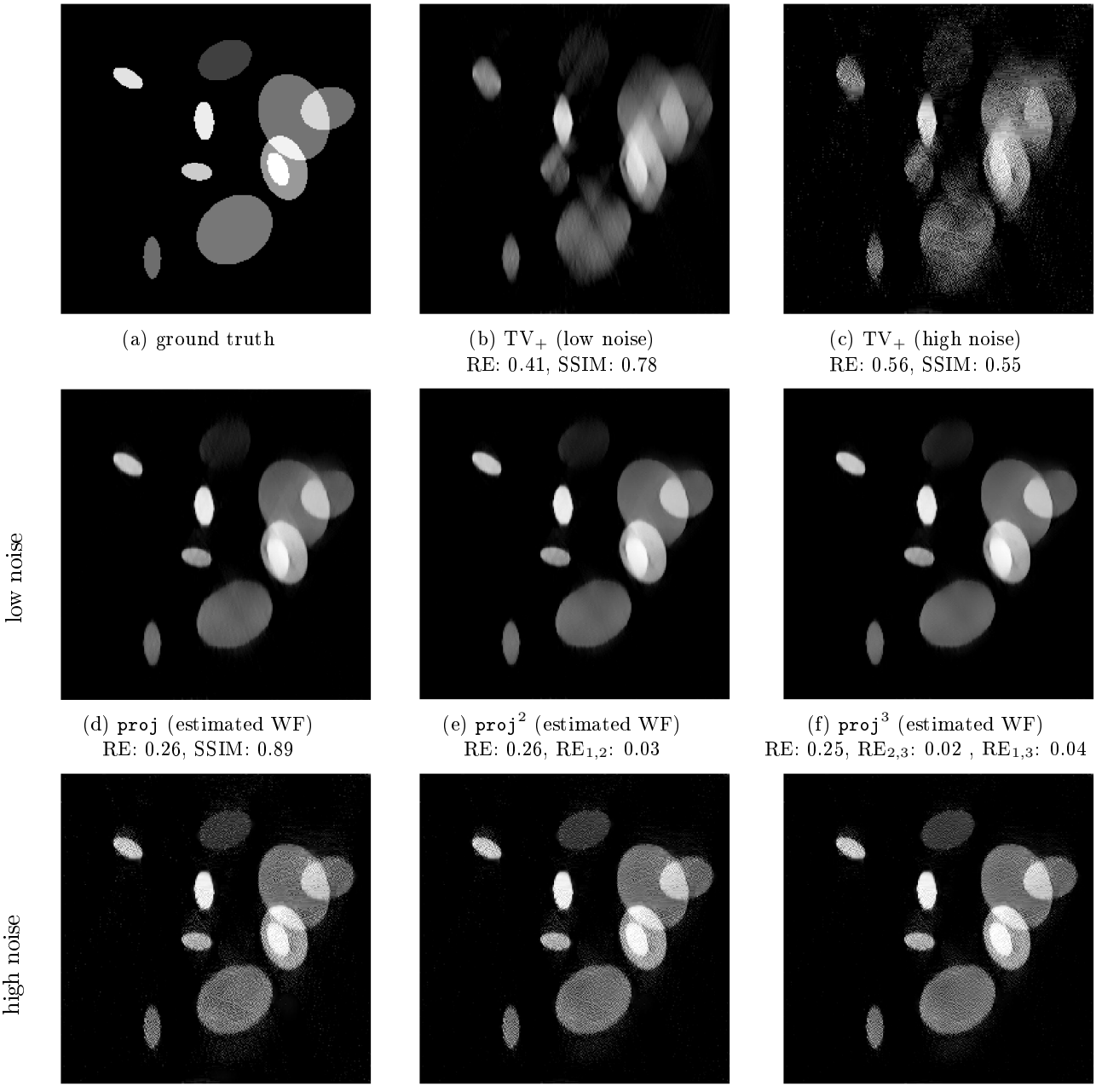
(f) `proj` (given WF)
RE: 0.20, SSIM: 0.90

(g) TV_+
RE: 0.60, SSIM: 0.61

(h) `proj` (estimated WF)
RE: 0.40, SSIM: 0.68

(i) `proj` (given WF)
RE: 0.26, SSIM: 0.72

Figure 21: Visualization of the results for one test sample in scenarios Ellipse-25° (low noise) and Ellipse-25° (high noise). For averaged similarity measures over the test set, see table 4.



(a) ground truth

(b) TV_+ (low noise)
RE: 0.41, SSIM: 0.78

(c) TV_+ (high noise)
RE: 0.56, SSIM: 0.55

low noise

(d) `proj` (estimated WF)
RE: 0.26, SSIM: 0.89

(e) $\texttt{proj}^2$ (estimated WF)
RE: 0.26, $\mathrm{RE}_{1,2}$: 0.03

(f) $\texttt{proj}^3$ (estimated WF)
RE: 0.25, $\mathrm{RE}_{2,3}$: 0.02 , $\mathrm{RE}_{1,3}$: 0.04

high noise

(g) `proj` (given WF)
RE: 0.33, SSIM: 0.66

(h) $\texttt{proj}^2$ (given WF)
RE: 0.31, $\mathrm{RE}_{1,2}$: 0.05

(i) $\texttt{proj}^3$ (given WF)
RE: 0.29, $\mathrm{RE}_{2,3}$: 0.03, $\mathrm{RE}_{1,3}$: 0.08

Figure 22: Visualization of the idempotency property of the projector network for one test sample in scenarios Ellipse-25° (low noise) and Ellipse-25° (high noise). Corresponding statistics are provided in table 5.

error of $\texttt{proj}^j$ w.r.t. $\texttt{proj}^i$. Visualization for this is presented in figure 22. These results show that the

error in idempotency between consecutive projections is only few percent and although it is larger after projecting three times the quality of reconstruction seems to even improve little when projecting more.

	high noise 1			high noise 2		
Method	**RE**	**SSIM**	**HaarPSI**	**RE**	**SSIM**	**HaarPSI**
`proj`	0.42 ± 0.07	$\mathbf{0.66 \pm 0.10}$	$\mathbf{0.31 \pm 0.05}$	$\mathbf{0.38 \pm 0.07}$	0.62 ± 0.08	$\mathbf{0.31 \pm 0.05}$
`proj oracle`	0.28 ± 0.06	0.70 ± 0.10	0.50 ± 0.06	0.21 ± 0.03	0.66 ± 0.07	0.54 ± 0.06

Table 3: These statistics are for comparing the performance of the projector networks trained with data Ellipse-25° (high noise 1 refers to this) and Ellipse-25° (high noise). Statistics for both networks are computed for Ellipse-25° (high noise) test set.

	low noise			high noise		
Method	**RE**	**SSIM**	**HaarPSI**	**RE**	**SSIM**	**HaarPSI**
FBP_+	1.01 ± 0.09	0.51 ± 0.06	0.13 ± 0.02	1.06 ± 0.14	0.31 ± 0.05	0.13 ± 0.02
TV_+	0.42 ± 0.07	0.75 ± 0.07	0.23 ± 0.03	0.52 ± 0.08	0.58 ± 0.10	0.20 ± 0.03
`proj`	$\mathbf{0.27 \pm 0.05}$	$\mathbf{0.88 \pm 0.04}$	$\mathbf{0.43 \pm 0.05}$	$\mathbf{0.42 \pm 0.07}$	$\mathbf{0.66 \pm 0.10}$	$\mathbf{0.31 \pm 0.05}$
`proj oracle`	0.17 ± 0.03	0.92 ± 0.03	0.61 ± 0.05	0.28 ± 0.06	0.70 ± 0.10	0.50 ± 0.06

Table 4: Statistics comparing the performance for Ellipse-25° for different noise levels. The low noise part is the same as $\mathcal{R}_{25^\circ}$ in the table 1 but it is repeated here for easier comparison.

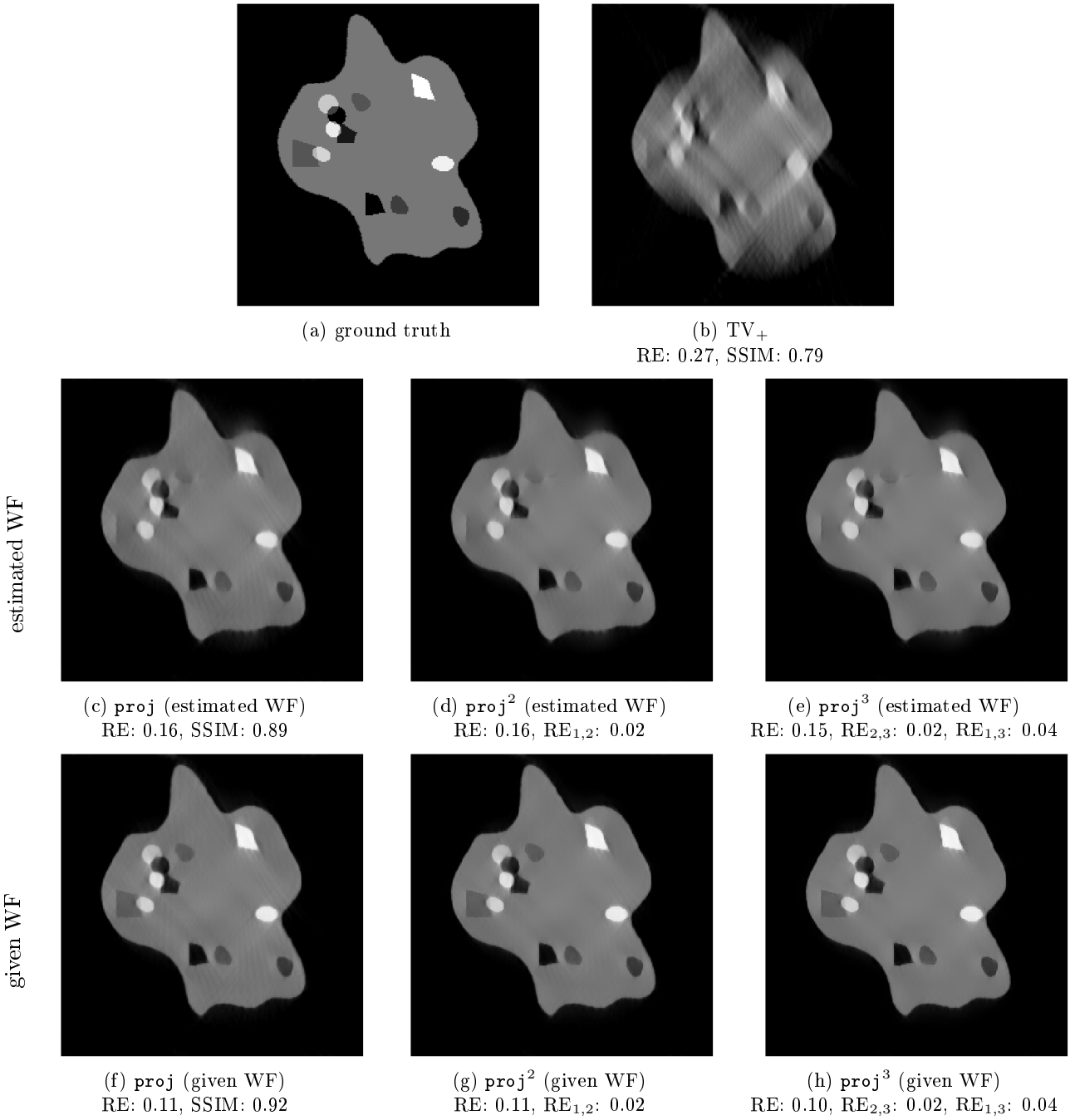
noise	**WF**	$\mathbf{RE}_{\texttt{proj}}$	$\mathbf{RE}_{\texttt{proj}^2}$	$\mathbf{RE}_{\texttt{proj}^3}$	$\mathbf{RE}_{1,2}$	$\mathbf{RE}_{2,3}$	$\mathbf{RE}_{1,3}$
low	estimated	0.27 ± 0.05	0.26 ± 0.05	0.26 ± 0.05	0.03 ± 0.01	0.02 ± 0.01	0.05 ± 0.02
	given	0.17 ± 0.03	0.16 ± 0.03	0.15 ± 0.03	0.02 ± 0.00	0.01 ± 0.00	0.03 ± 0.01
high	estimated	0.42 ± 0.07	0.42 ± 0.07	0.42 ± 0.08	0.09 ± 0.03	0.06 ± 0.02	0.14 ± 0.05
	given	0.28 ± 0.06	0.26 ± 0.05	0.24 ± 0.05	0.04 ± 0.01	0.03 ± 0.01	0.07 ± 0.02

Table 5: Statistics for assessing idempotency property on test sets of Ellipse-25° with different noise levels.

Method	**RE**	**SSIM**	**HaarPSI**
TV_+	0.29 ± 0.04	0.79 ± 0.04	0.29 ± 0.04
`proj`	$\mathbf{0.19 \pm 0.03}$	$\mathbf{0.87 \pm 0.04}$	$\mathbf{0.50 \pm 0.07}$
`proj oracle`	0.12 ± 0.01	0.92 ± 0.02	0.71 ± 0.06

Table 6: Performance statistics for the test set of Smooth-40° scenario. Example reconstructions are shown in figure 23.

Table 7 and figure 23 shows similar results for the Smooth-40° scenario. Moreover, figure 23 with figure 25 and table 6 presents performance of the proposed method on Smooth-40° scenario. Finally figure 24



(a) ground truth

(b) TV_+
RE: 0.27, SSIM: 0.79

(c) `proj` (estimated WF)
RE: 0.16, SSIM: 0.89

(d) $\texttt{proj}^2$ (estimated WF)
RE: 0.16, $\mathrm{RE}_{1,2}$: 0.02

(e) $\texttt{proj}^3$ (estimated WF)
RE: 0.15, $\mathrm{RE}_{2,3}$: 0.02, $\mathrm{RE}_{1,3}$: 0.04

(f) `proj` (given WF)
RE: 0.11, SSIM: 0.92

(g) $\texttt{proj}^2$ (given WF)
RE: 0.11, $\mathrm{RE}_{1,2}$: 0.02

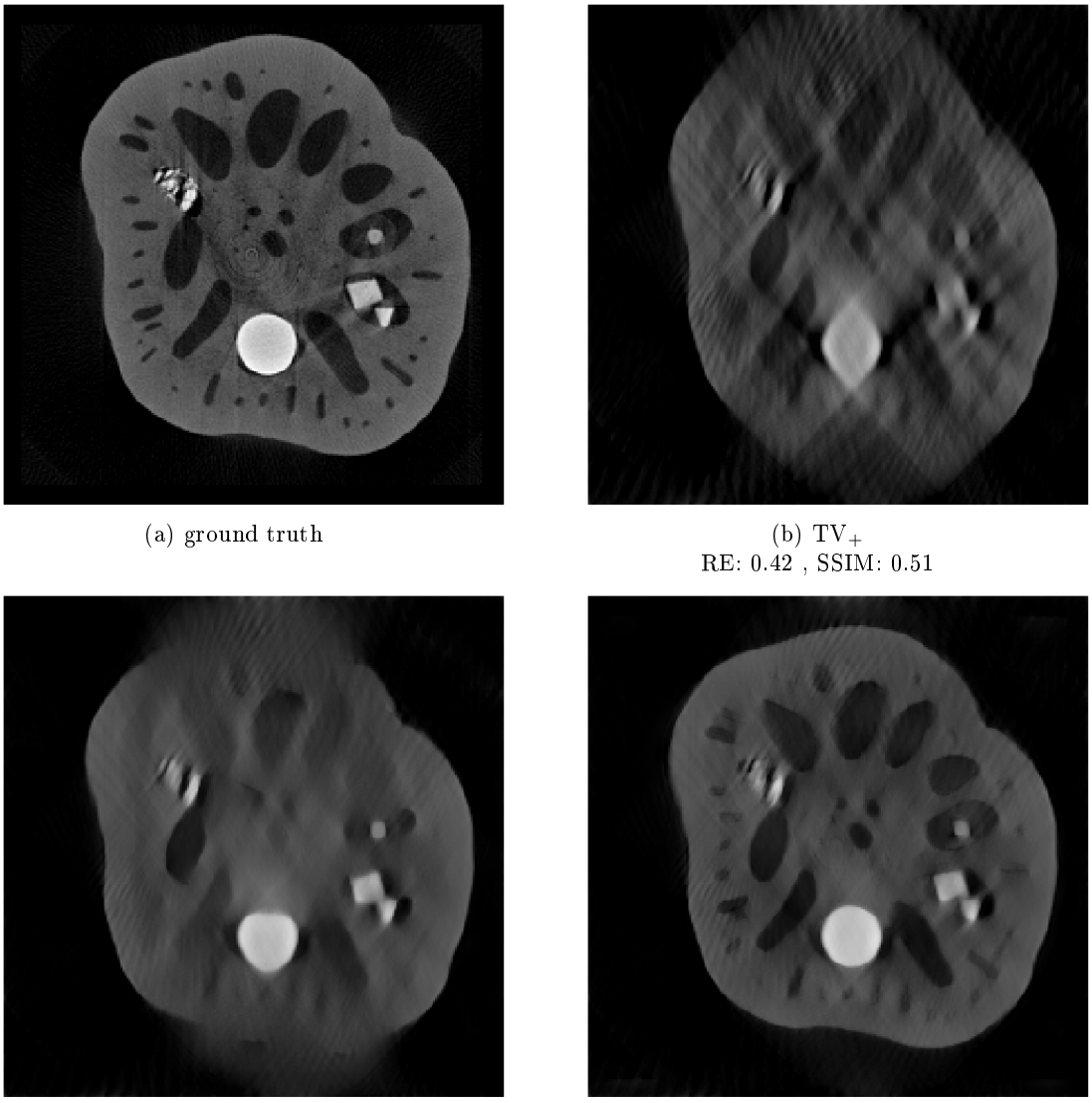
(h) $\texttt{proj}^3$ (given WF)
RE: 0.10, $\mathrm{RE}_{2,3}$: 0.02, $\mathrm{RE}_{1,3}$: 0.04

Figure 23: Visualization of the reconstructions and the idempotency property of the projector network for one test sample in Smooth-40°. Corresponding statistics are provided in tables 6 and 7. Figure 25 shows a visualization of the wavefront set estimation for this test sample.

provides example reconstructions for the real measured data. The lotus root is a more complex target than

WF	$\mathbf{RE}_{\mathtt{proj}}$	$\mathbf{RE}_{\mathtt{proj}^2}$	$\mathbf{RE}_{\mathtt{proj}^3}$	$\mathbf{RE}_{1,2}$	$\mathbf{RE}_{2,3}$	$\mathbf{RE}_{1,3}$
estimated	0.19 ± 0.03	0.19 ± 0.03	0.19 ± 0.04	0.03 ± 0.01	0.02 ± 0.01	0.05 ± 0.01
given	0.12 ± 0.01	0.11 ± 0.01	0.11 ± 0.02	0.02 ± 0.00	0.02 ± 0.00	0.04 ± 0.01

Table 7: Statistics for assessing idempotency property on Smooth-40° test set.



(a) ground truth

(b) TV_+
RE: 0.42 , SSIM: 0.51

(c) `proj` (estimated)
RE: 0.38, SSIM: 0.53

(d) `proj` (given WF)
RE: 0.28, SSIM: 0.63

Figure 24: Visualization of the performance on lotus root data from real $\mathcal{R}_{40^\circ}$ fanbeam measurement.

the ones simulated in the smooth data and the projector network with an estimated wavefront set does not improve the quality of the reconstruction significantly. However, when the true wavefront set is given, the achieved reconstruction is much better, which suggests the projector network seems to generalize quite well.

(a) $\mathcal{R}_{40^\circ}$ input
ℓ^1-RE: 1.01

(b) $\mathcal{R}_{40^\circ}$ output
ℓ^1-RE: 0.77

(c) ground truth

Figure 25: Visualizations of the wavefront set estimation for the scenario Smooth-40°. For the input in subfigure a, error is computed only for the visible parts of the wavefront set. Statistics for the wavefront set estimation are provided in table 2.

6 Discussion

This thesis has presented a method for solving very limited angle tomography problems. The method uses two U-Net like neural networks to improve a model-based reconstruction. One network to infer the invisible wavefront set from the visible one and another to project the model-based reconstruction such that the projection has the estimated wavefront set. Numerical experiments show that the quality of the model-based reconstructions was improved notably. However, the performance of the method might improve by further studying, since several options were not studied in this thesis. This is the case because training neural networks requires quite a considerable amount of computation time. These options are shortly discussed in this section.

The used training data sets were quite small for deep learning. They required still a large amount of memory since one training example contained a pair of shearlet transforms of size $256 \times 256 \times 49$ (twice 3211264 elements). The memory requirement could be reduced by saving inputs and outputs in the image domain instead of the shearlet domain, and computing shearlet transforms and inverse transforms during the training. This was not done, because the computation of the shearlet transform is relatively slow. However, presenting outputs in the image domain during the training has some benefits for the method. It allows using non-negativity constraint already during the training in contrast to training in the shearlet domain, which probably improves the reconstructions to some extent. Moreover, computing the loss function in the image domain could be more reasonable since good performance in the image domain is the primal goal. This more memory efficient choice would be useful for changing the resolution of the targets for more common 512×512 instead of used 256×256. Note that these changes would increase the heaviness of computations and require more RAM for GPU.

There are also many choices related to inputs of neural networks to study. It would be interesting to see how the proposed method would perform if trained with real (medical) data. One simple change to the method is to train the projector network using the estimated wavefront sets in inputs instead of the true ones. Training of both networks could be also done simultaneously instead of one at a time. There is also option to replace used wavefront set representation with some other. The current representation could be transformed from part of the cleaned shearlet coefficients to the wavefront set. One simple approach for this transformation is to use some threshold to decide that a point in the shearlet domain belongs to the wavefront set if and only is its value of the cleaned shearlet transform is greater than the threshold. The article [2] presents one interesting way for wavefront set extraction using shearlets and deep learning. Improving the quality of model-based reconstruction, primal input of the method, could affect the performance more than improvements for wavefront set reconstruction. The method proposed in the article [8] put more effort into the model-based reconstructions and maybe a similar approach could be useful for this method.

Better performance could be achieved developing iterative algorithm using the idea of projecting reconstruction such that the projection has the desired wavefront set. The assessing of the idempotency property in section 5.4 shows that the quality of reconstruction results in a little improvement when projected multiple times. This is promising for the development of the iterative algorithm.

References

[1] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.

[2] Héctor Andrade-Loarca, Gitta Kutyniok, Ozan Öktem, and Philipp Petersen. Extraction of digital wavefront sets using applied harmonic analysis and deep neural networks. *arXiv preprint arXiv:1901.01388*, 2019.

[3] Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.

[4] Jacques Arsac. *Fourier transforms and the theory of distributions*. Prentice Hall, 1966.

[5] Salomon Bochner, Komaravolu Chandrasekharan, et al. *Fourier transforms*. Princeton University Press, 1949.

[6] Christian Brouder, Nguyen Viet Dang, and Frédéric Hélein. A smooth introduction to the wavefront set. *Journal of Physics A: Mathematical and Theoretical*, 47(44):443001, 2014.

[7] Tatiana A Bubba, Andreas Hauptmann, Simo Huotari, Juho Rimpeläinen, and Samuli Siltanen. Tomographic X-ray data of a lotus root filled with attenuating objects. *arXiv preprint arXiv:1609.07299*, 2016.

[8] Tatiana A Bubba, Gitta Kutyniok, Matti Lassas, Maximilian März, Wojciech Samek, Samuli Siltanen, and Vignesh Srinivasan. Learning the invisible: A hybrid deep learning-shearlet framework for limited angle computed tomography. *Inverse Problems*, 35(6):064002, 2019.

[9] Peijun Chen, Jianguo Huang, and Xiaoqun Zhang. A primal-dual fixed point algorithm for minimization of the sum of three convex separable functions. *Fixed Point Theory and Applications*, 2016(1):54, 2016.

[10] Allan Macleod Cormack. Representation of a function by its line integrals, with some radiological applications. *Journal of applied physics*, 34(9):2722–2727, 1963.

[11] National Research Council et al. *Mathematics and physics of emerging biomedical imaging*. National Academies Press, 1996.

[12] Ingrid Daubechies. *Ten lectures on wavelets*, volume 61. Siam, 1992.

[13] Mark E Davison. The ill-conditioned nature of the limited angle tomography problem. *SIAM Journal on Applied Mathematics*, 43(2):428–448, 1983.

[14] Leonardo De Chiffre, Simone Carmignato, J-P Kruth, Robert Schmitt, and Albert Weckenmann. Industrial applications of computed tomography. *CIRP annals*, 63(2):655–677, 2014.

[15] Eleonora Di Nezza, Giampiero Palatucci, and Enrico Valdinoci. Hitchhikers guide to the fractional Sobolev spaces. *Bulletin des Sciences Mathématiques*, 136(5):521–573, 2012.

[16] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[17] Jürgen Frikel and Eric Todd Quinto. Characterization and reduction of artifacts in limited angle tomography. *Inverse Problems*, 29(12):125007, 2013.

[18] Qichuan Geng, Zhong Zhou, and Xiaochun Cao. Survey of recent progress in semantic image segmentation with CNNs. *Science China Information Sciences*, 61(5):051101, 2018.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[20] Loukas Grafakos. *Classical fourier analysis*, volume 2. Springer, 2008.

[21] Philipp Grohs. Continuous shearlet frames and resolution of the wavefront set. *Monatshefte für Mathematik*, 164(4):393–426, 2011.

[22] Jawook Gu and Jong Chul Ye. Multi-scale wavelet domain residual learning for limited-angle CT reconstruction. *arXiv preprint arXiv:1703.01382*, 2017.

[23] Yoseob Han and Jong Chul Ye. Framing U-Net via deep convolutional framelets: Application to sparse-view CT. *IEEE transactions on medical imaging*, 37(6):1418–1429, 2018.

[24] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[25] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[26] Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.

[27] Emily J King, Rafael Reisenhofer, Johannes Kiefer, Wang-Q Lim, Zhen Li, and Georg Heygster. Shearlet-based edge detection: flame fronts and tidal flats. In *Applications of Digital Image Processing XXXVIII*, volume 9599, page 959905. International Society for Optics and Photonics, 2015.

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[29] Erwin Kreyszig. *Introductory functional analysis with applications*, volume 1. wiley New York, 1978.

[30] Venkateswaran P Krishnan and Eric Todd Quinto. Microlocal analysis in tomography. *Handbook of mathematical methods in imaging*, pages 1–50, 2014.

[31] Gitta Kutyniok and Demetrio Labate. *Shearlets: Multiscale analysis for multivariate data*. Springer Science & Business Media, 2012.

[32] Gitta Kutyniok, Wang-Q Lim, and Rafael Reisenhofer. Shearlab 3D: Faithful digital shearlet transforms based on compactly supported shearlets. *arXiv preprint arXiv:1402.5670*, 2014.

[33] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[34] Tim Meinhardt, Michael Moller, Caner Hazirbas, and Daniel Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1781–1790, 2017.

[35] Thomas Mertelmeier, Jasmina Ludwig, Bo Zhao, and Wei Zhao. Optimization of tomosynthesis acquisition parameters: angular range and number of projections. In *International Workshop on Digital Mammography*, pages 220–227. Springer, 2008.

[36] Nikita Moriakov, Koen Michielsen, Jonas Adler, Ritse Mann, Ioannis Sechopoulos, and Jonas Teuwen. Deep learning framework for digital breast tomosynthesis reconstruction. In *Medical Imaging 2019: Physics of Medical Imaging*, volume 10948, page 1094804. International Society for Optics and Photonics, 2019.

[37] Jennifer L Mueller and Samuli Siltanen. *Linear and nonlinear inverse problems with practical applications*, volume 10. Siam, 2012.

[38] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

[39] Eric Todd Quinto. Singularities of the X-ray transform and limited data tomography in $\mathbb{R}^2$ and $\mathbb{R}^3$. *SIAM Journal on Mathematical Analysis*, 24(5):1215–1225, 1993.

[40] Rafael Reisenhofer, Sebastian Bosse, Gitta Kutyniok, and Thomas Wiegand. A haar wavelet-based perceptual similarity index for image quality assessment. *Signal Processing: Image Communication*, 61:33–43, 2018.

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[42] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by back-propagating errors in parallel distributed processing: Explorations in the microstructure of cognition. eds, 1986.

[43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[44] Thomas W Sederberg. Computer aided geometric design. 2012.

[45] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[46] Hanming Zhang, Liang Li, Kai Qiao, Linyuan Wang, Bin Yan, Lei Li, and Guoen Hu. Image prediction for limited-angle tomography via deep learning with convolutional neural network. *arXiv preprint arXiv:1607.08707*, 2016.

[47] Yiheng Zhang, Heang-Ping Chan, Berkman Sahiner, Jun Wei, Mitchell M Goodsitt, Lubomir M Hadjiiski, Jun Ge, and Chuan Zhou. A comparative study of limited-angle cone-beam reconstruction methods for breast tomosynthesis. *Medical physics*, 33(10):3781–3795, 2006.